

# PATENT ABSTRACTS OF JAPAN

**(11)Publication number : 2001-060150**

**(43)Date of publication of application : 06.03.2001**

(51)Int.Cl.

**G06F 9/06**

**G06F 12/00**

**G06F 17/30**

(21)Application number : **11-233910**

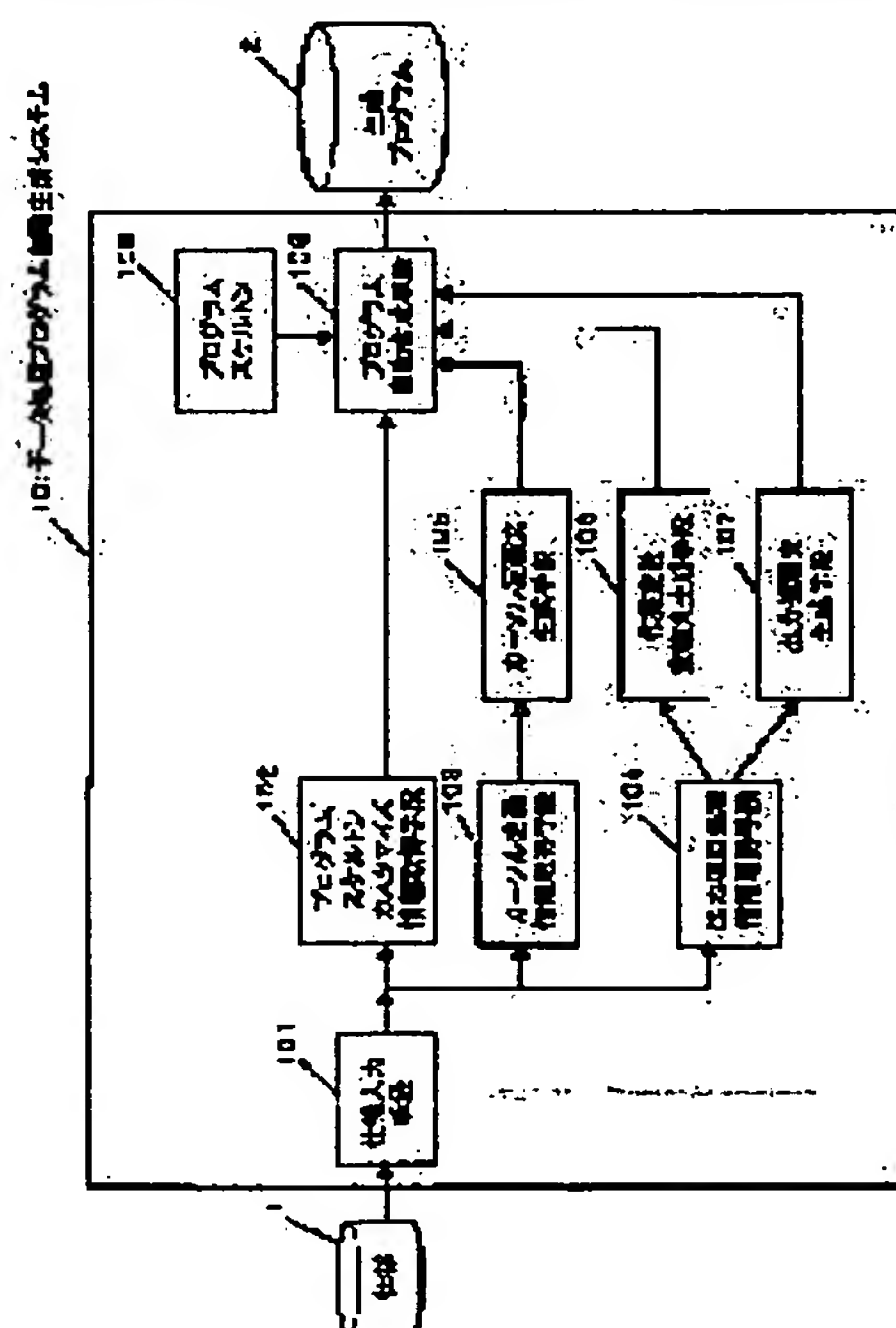
(71)Applicant : **NEC CORP**

(22)Date of filing : **20.08.1999**

(72)Inventor : YAMANOUCHI TORU

**SATO AKIRA**

**(54) AUTOMATIC DATA PROCESSING PROGRAM GENERATION SYSTEM, METHOD THEREFOR AND COMPUTER READABLE RECORDING MEDIUM**



**(57)Abstract:**

**PROBLEM TO BE SOLVED:** To automatically generate a data processing program for referring to, updating and eliminating the chart data of a data base by inserting generated cursor definition sentence and work variable definition sentence, etc., to the pertinent parts of a program skeleton.

**SOLUTION:** An automatic program synthesis means 109 inserts a program name obtained in a program skeleton customize information obtaining means 102 to an insertion position corresponding to the program name among the insertion positions of the program skeleton 108. Also, the cursor definition sentence generated in a cursor definition sentence generation means 105 is inserted. Then, the work variable

definition sentence generated in a work variable definition sentence generation means 106 is inserted to the insertion position corresponding to the work variable definition sentence, an output processing sentence generated in an output processing sentence generation means 107 is inserted to the insertion

**BEST AVAILABLE COPY**

position corresponding to the output processing sentence, the fragments of a corresponding program sentence obtained in the program skeleton customize information obtaining means 102 are inserted to the other insertion positions, and thus, the program is automatically synthesized.

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2001-60150

(P2001-60150A)

(43) 公開日 平成13年 3 月 6 日 (2001. 3. 6)

(51) Int.Cl. <sup>7</sup>	識別記号	F I	テ-マコ-ト <sup>*</sup> (参考)
G 0 6 F 9/06	5 3 0	G 0 6 F 9/06	5 3 0 V 5 B 0 7 5
12/00	5 1 3	12/00	5 1 3 Z 5 B 0 7 6
17/30		15/403	3 4 0 D 5 B 0 8 2

審査請求 有 請求項の数10 O L (全 32 頁)

(21) 出願番号 特願平11-233910

(22) 出願日 平成11年 8 月 20 日 (1999. 8. 20)

(71) 出願人 000004237

日本電気株式会社

東京都港区芝五丁目 7 番 1 号

(72) 発明者 山之内 徹

東京都港区芝五丁目 7 番 1 号 日本電気株式会社内

(72) 発明者 佐藤 明良

東京都港区芝五丁目 7 番 1 号 日本電気株式会社内

(74) 代理人 100088959

弁理士 境 廣巳

Fターム(参考) 5B075 ND02 QR03 QT06

5B076 DD04

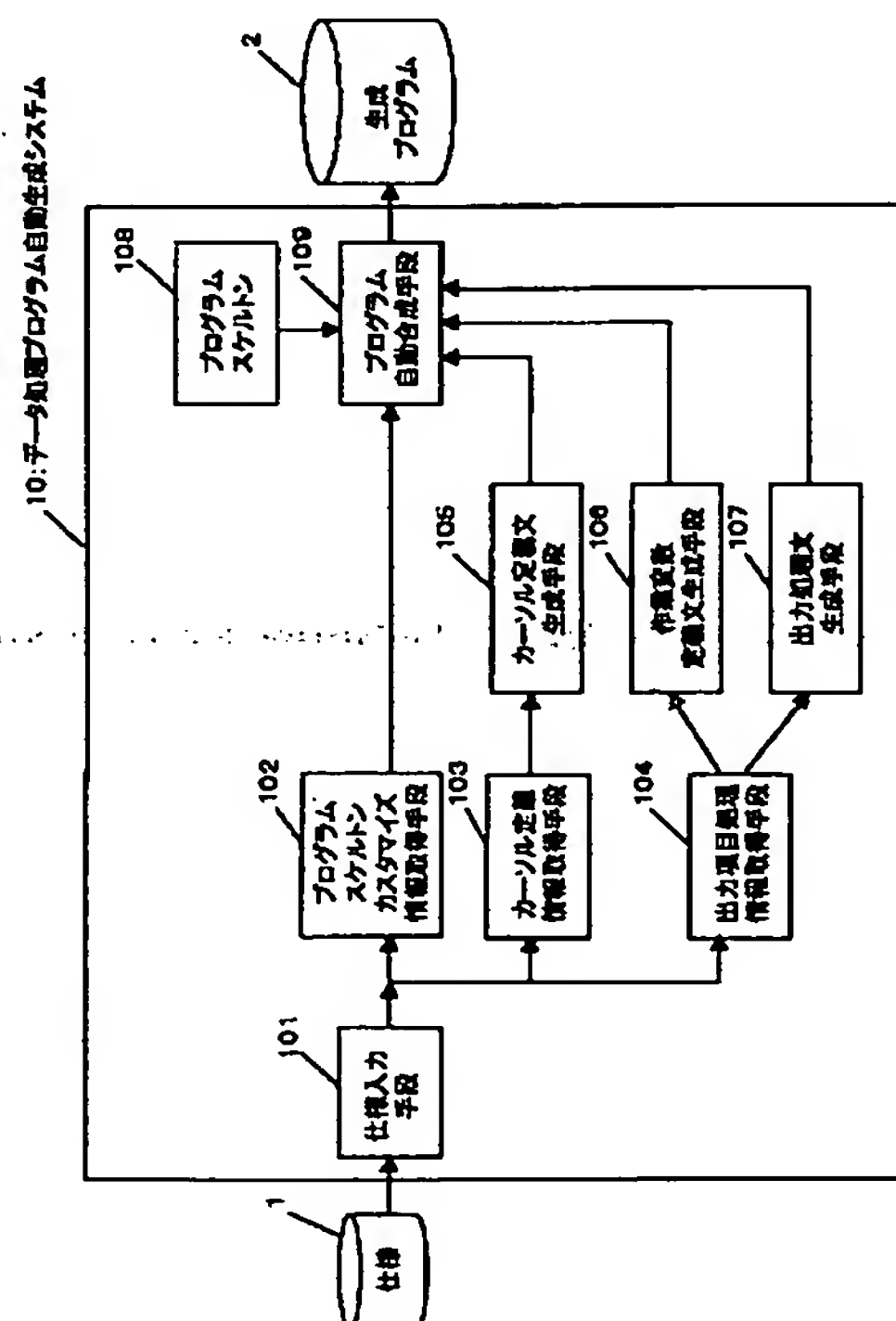
5B082 GA07

(54) 【発明の名称】 データ処理プログラム自動生成システム及びその方法並びにコンピュータ可読記録媒体

(57) 【要約】

【課題】 データベースの表のデータをカーソル定義文を使って参照、更新、削除するデータ処理プログラムを自動的に生成する。

【解決手段】 手段101 は仕様 1 を入力する。仕様 1 から、手段102 はプログラムカスタマイズ情報を、手段103 は入力表項目定義とセレクト項目定義とセレクト条件定義を、手段104 は出力表項目定義と出力表処理定義を、それぞれ取り出す。手段105 は手段103 で取得された情報からカーソル定義文を、手段106、107は手段104 で取得された情報から作業変数定義文、指定された処理種別 (挿入、更新、削除) に応じた出力処理文を、それぞれ生成する。手段109 は、仕様の如何に関わらず固定の部分のプログラムと仕様によって変化する部分のプログラム断片の挿入位置とを含むプログラムスケルトン108 の該当挿入位置に、プログラムカスタマイズ情報とカーソル定義文と作業変数定義文と出力処理文を挿入してプログラム 2 を生成する。



## 【特許請求の範囲】

【請求項 1】 データベースの表のデータを参照、更新、削除するデータ処理プログラムを自動的に生成するデータ処理プログラム自動生成システムであって、仕様から取得した情報に基づいて、カーソル定義文、作業変数定義文、出力処理文を生成する手段と、前記データ処理プログラムのうち、仕様の如何に関わらず固定の部分のプログラムと仕様によって変化する部分のプログラム断片の挿入位置とを含むプログラムスケルトンの該当箇所に、前記生成されたカーソル定義文、作業変数定義文、出力処理文、および前記仕様中のプログラムカスタマイズ情報を挿入して前記データ処理プログラムを完成させ、出力する手段とを含むデータ処理プログラム自動生成システム。

【請求項 2】 データベースの表のデータを参照、更新、削除するデータ処理プログラムを自動的に生成するデータ処理プログラム自動生成システムであって、プログラム名と入力表項目定義とセレクト項目定義と出力表項目定義と出力表処理定義とセレクト条件定義とプログラムカスタマイズ情報とを含む仕様を入力する仕様入力手段と、

前記仕様入力手段で入力された仕様からプログラムスケルトンをカスタマイズするための情報として、プログラム名とプログラムカスタマイズ情報を取り出すプログラムスケルトンカスタマイズ情報取得手段と、

前記仕様入力手段で入力された仕様から入力表項目定義とセレクト項目定義とセレクト条件定義を取り出すカーソル定義情報取得手段と、

前記仕様入力手段で入力された仕様から出力表項目定義と出力表処理定義を取り出す出力項目処理情報取得手段と、

前記カーソル定義情報取得手段で取得された情報からカーソル定義文を生成するカーソル定義文生成手段と、

前記出力項目処理情報取得手段で取得された出力表項目定義のうち、処理種別が挿入である出力表に対応するものから作業変数定義文を生成する作業変数定義文生成手段と、

前記出力項目処理情報取得手段で取得された出力表項目定義と出力表処理定義から、処理種別が挿入の場合は処理条件文と項目計算文と挿入文とを含む出力処理文を生成し、処理種別が更新の場合には処理条件文と更新文とを含む出力処理文を生成し、処理種別が削除の場合には処理条件文と削除文とを含む出力処理文を生成する出力処理文生成手段と、

前記データ処理プログラムのうち、仕様の如何に関わらず固定の部分のプログラムと仕様によって変化する部分のプログラム断片の挿入位置とを含むプログラムスケルトンの該当挿入位置に、前記プログラムスケルトンカスタマイズ情報取得手段で取得したプログラム名とプログラムカスタマイズ情報と、前記カーソル定義文生成手段

で生成したカーソル定義文と、前記作業変数定義文生成手段で生成した作業変数定義文と、前記出力処理文生成手段で生成した出力処理文とを挿入することによってプログラムを完成させ、出力するプログラム自動合成手段とを有するデータ処理プログラム自動生成システム。

【請求項 3】 データベースの表のデータを参照、更新、削除するデータ処理プログラムを自動的に生成するデータ処理プログラム自動生成システムであって、プログラム名と入力表項目定義と出力表項目定義と出力表処理定義とセレクト条件定義とプログラムカスタマイズ情報とを含む仕様を入力する仕様入力手段と、

前記仕様入力手段で入力された仕様からプログラムスケルトンをカスタマイズするための情報として、プログラム名とプログラムカスタマイズ情報を取り出すプログラムスケルトンカスタマイズ情報取得手段と、

前記仕様入力手段で入力された仕様から入力表項目定義とセレクト条件定義と出力表処理定義とを取り出すカーソル定義情報取得手段と、

前記仕様入力手段で入力された仕様から出力表項目定義と出力表処理定義を取り出す出力項目処理情報取得手段と、

前記カーソル定義情報取得手段で取得された入力表項目定義とセレクト条件定義と出力表処理定義とからセレクト項目定義を作成し、該作成したセレクト項目定義と前記カーソル定義情報取得手段で取得されたセレクト条件定義とからカーソル定義文を生成するカーソル定義文生成手段と、

前記出力項目処理情報取得手段で取得された出力表項目定義のうち、処理種別が挿入である出力表に対応するものから作業変数定義文を生成する作業変数定義文生成手段と、

前記出力項目処理情報取得手段で取得された出力表項目定義と出力表処理定義から、処理種別が挿入の場合は処理条件文と項目計算文と挿入文とを含む出力処理文を生成し、処理種別が更新の場合には処理条件文と更新文とを含む出力処理文を生成し、処理種別が削除の場合には処理条件文と削除文とを含む出力処理文を生成する出力処理文生成手段と、

前記データ処理プログラムのうち、仕様の如何に関わらず固定の部分のプログラムと仕様によって変化する部分のプログラム断片の挿入位置とを含むプログラムスケルトンの該当挿入位置に、前記プログラムスケルトンカスタマイズ情報取得手段で取得したプログラム名とプログラムカスタマイズ情報と、前記カーソル定義文生成手段で生成したカーソル定義文と、前記作業変数定義文生成手段で生成した作業変数定義文と、前記出力処理文生成手段で生成した出力処理文とを挿入することによってプログラムを完成させ、出力するプログラム自動合成手段とを有するデータ処理プログラム自動生成システム。

【請求項 4】 仕様の如何に関わらず固定の部分のプロ



グラムと仕様によって変化する部分のプログラム断片の挿入位置とを含むプログラムスケルトンを複数種類格納するプログラムスケルトンライブラリと、

前記プログラムスケルトンライブラリから利用者の指定した1つのプログラムスケルトンを選択し、前記プログラム自動合成手段に渡すプログラムスケルトン選択手段とを備えた請求項2または3記載のデータ処理プログラム自動生成システム。

【請求項5】 前記仕様入力手段は、処理で使用する全ての入力表名と出力表名を入力し、データベースの表定義情報から当該入力表と出力表の全ての項目名と型情報を検索して取得すると共に、これを画面上に提示して仕様記述者による入力表項目定義および出力表項目定義以外の仕様情報の指定を可能とする構成を有する請求項2、3または4記載のデータ処理プログラム自動生成システム。

【請求項6】 前記プログラムスケルトンはデバッグ用トレース文を含み、

前記プログラム自動合成手段は、デバッグ用のプログラムを生成するのか、本番用のプログラムを生成するのかの別を示す生成プログラム識別情報に従って、前記プログラムスケルトン中のデバッグ用トレース文を取り去るか、残すかを切りわけ構成を有する請求項2、3、4または5記載のデータ処理プログラム自動生成システム。

【請求項7】 データベースの表のデータを参照、更新、削除するデータ処理プログラムを自動的に生成するデータ処理プログラム自動生成方法であって、(a) プログラム名と入力表項目定義とセレクト項目定義と出力表項目定義と出力表処理定義とセレクト条件定義とプログラムカスタマイズ情報とを含む仕様を入力するステップと、(b) ステップaで入力された仕様からプログラムスケルトンをカスタマイズするための情報として、プログラム名とプログラムカスタマイズ情報を取り出すステップと、(c) ステップaで入力された仕様から入力表項目定義とセレクト項目定義とセレクト条件定義を取り出すステップと、(d) ステップaで入力された仕様から出力表項目定義と出力表処理定義を取り出すステップと、(e) ステップcで取得された情報からカーソル定義文を生成するステップと、(f) ステップdで取得された出力表項目定義のうち、処理種別が挿入である出力表に対応するものから作業変数定義文を生成するステップと、(g) ステップdで取得された出力表項目定義と出力表処理定義から、処理種別が挿入の場合は処理条件文と項目計算文と挿入文とを含む出力処理文を生成し、処理種別が更新の場合には処理条件文と更新文とを含む出力処理文を生成し、処理種別が削除の場合には処理条件文と削除文とを含む出力処理文を生成するステップと、(h) 前記データ処理プログラムのうち、仕様の如何に関わらず固定の部分のプログラムと仕様によって変化する部分のプログラム断片の挿入位置とを含むプロ

グラムスケルトンの該当挿入位置に、ステップbで取得したプログラム名とプログラムカスタマイズ情報と、前記生成したカーソル定義文と、前記生成した作業変数定義文と、前記生成した出力処理文とを挿入することによってプログラムを完成させ、出力するプログラムステップとを含むデータ処理プログラム自動生成方法。

【請求項8】 データベースの表のデータを参照、更新、削除するデータ処理プログラムを自動的に生成するデータ処理プログラム自動生成方法であって、(a) プログラム名と入力表項目定義と出力表項目定義と出力表処理定義とセレクト条件定義とプログラムカスタマイズ情報とを含む仕様を入力するステップと、(b) ステップaで入力された仕様からプログラムスケルトンをカスタマイズするための情報として、プログラム名とプログラムカスタマイズ情報を取り出すステップと、(c) ステップaで入力された仕様から入力表項目定義とセレクト条件定義と出力表処理定義とを取り出すステップと、(d) ステップaで入力された仕様から出力表項目定義と出力表処理定義を取り出すステップと、(e) ステップcで取得された入力表項目定義とセレクト条件定義と出力表処理定義とからセレクト項目定義を作成し、該作成したセレクト項目定義と前記セレクト条件定義とからカーソル定義文を生成するステップと、(f) ステップdで取得された出力表項目定義のうち、処理種別が挿入である出力表に対応するものから作業変数定義文を生成するステップと、(g) ステップdで取得された出力表項目定義と出力表処理定義から、処理種別が挿入の場合は処理条件文と項目計算文と挿入文とを含む出力処理文を生成し、処理種別が更新の場合には処理条件文と更新文とを含む出力処理文を生成し、処理種別が削除の場合には処理条件文と削除文とを含む出力処理文を生成するステップと、(h) 前記データ処理プログラムのうち、仕様の如何に関わらず固定の部分のプログラムと仕様によって変化する部分のプログラム断片の挿入位置とを含むプログラムスケルトンの該当挿入位置に、ステップbで取得したプログラム名とプログラムカスタマイズ情報と、前記生成したカーソル定義文と、前記生成した作業変数定義文と、前記生成した出力処理文とを挿入することによってプログラムを完成させ、出力するステップとを含むデータ処理プログラム自動生成方法。

【請求項9】 データベースの表のデータを参照、更新、削除するデータ処理プログラムを自動的に生成するデータ処理プログラムを自動生成するプログラムを記録した媒体であって、コンピュータを、プログラム名と入力表項目定義とセレクト項目定義と出力表項目定義と出力表処理定義とセレクト条件定義とプログラムカスタマイズ情報とを含む仕様を入力する仕様入力手段、前記仕様入力手段で入力された仕様からプログラムスケルトンをカスタマイズするための情報として、プログラ

ム名とプログラムカスタマイズ情報を取り出すプログラム  
 スケルトンカスタマイズ情報取得手段、  
 前記仕様入力手段で入力された仕様から入力表項目定義  
 とセレクト項目定義とセレクト条件定義を取り出すカー  
 ソル定義情報取得手段、  
 前記仕様入力手段で入力された仕様から出力表項目定義  
 と出力表処理定義を取り出す出力項目処理情報取得手  
 段、  
 前記カーソル定義情報取得手段で取得された情報からカー  
 ソル定義文を生成するカーソル定義文生成手段、  
 前記出力項目処理情報取得手段で取得された出力表項目  
 定義のうち、処理種別が挿入である出力表に対応するも  
 のから作業変数定義文を生成する作業変数定義文生成手  
 段、  
 前記出力項目処理情報取得手段で取得された出力表項目  
 定義と出力表処理定義から、処理種別が挿入の場合は処  
 理条件文と項目計算文と挿入文とを含む出力処理文を生  
 成し、処理種別が更新の場合には処理条件文と更新文と  
 を含む出力処理文を生成し、処理種別が削除の場合には  
 処理条件文と削除文とを含む出力処理文を生成する出力  
 処理文生成手段、  
 前記データ処理プログラムのうち、仕様の如何に関わら  
 ず固定の部分のプログラムと仕様によって変化する部分  
 のプログラム断片の挿入位置とを含むプログラムスケル  
 トンの該当挿入位置に、前記プログラムスケルトンカス  
 タマイズ情報取得手段で取得したプログラム名とプログ  
 ラムカスタマイズ情報と、前記カーソル定義文生成手段  
 で生成したカーソル定義文と、前記作業変数定義文生成  
 手段で生成した作業変数定義文と、前記出力処理文生成  
 手段で生成した出力処理文とを挿入することによってプ  
 ログラムを完成させ、出力するプログラム自動合成手  
 段、  
 として機能させるプログラムを記録したコンピュータ可  
 読記録媒体。

【請求項 10】 データベースの表のデータを参照、更  
 新、削除するデータ処理プログラムを自動的に生成する  
 データ処理プログラムを自動生成するプログラムを記録  
 した媒体であって、コンピュータを、  
 プログラム名と入力表項目定義と出力表項目定義と出力  
 表処理定義とセレクト条件定義とプログラムカスタマイ  
 ズ情報とを含む仕様を入力する仕様入力手段、前記仕様  
 入力手段で入力された仕様からプログラムスケルトンを  
 カスタマイズするための情報として、プログラム名とプ  
 ログラムカスタマイズ情報を取り出すプログラムスケル  
 トンカスタマイズ情報取得手段、  
 前記仕様入力手段で入力された仕様から入力表項目定義  
 とセレクト条件定義と出力表処理定義とを取り出すカー  
 ソル定義情報取得手段、  
 前記仕様入力手段で入力された仕様から出力表項目定義  
 と出力表処理定義を取り出す出力項目処理情報取得手

段、  
 前記カーソル定義情報取得手段で取得された入力表項目  
 定義とセレクト条件定義と出力表処理定義とからセレクト  
 項目定義を作成し、該作成したセレクト項目定義と前  
 記カーソル定義情報取得手段で取得されたセレクト条件  
 定義とからカーソル定義文を生成するカーソル定義文生  
 成手段、  
 前記出力項目処理情報取得手段で取得された出力表項目  
 定義のうち、処理種別が挿入である出力表に対応するも  
 のから作業変数定義文を生成する作業変数定義文生成手  
 段、  
 前記出力項目処理情報取得手段で取得された出力表項目  
 定義と出力表処理定義から、処理種別が挿入の場合は処  
 理条件文と項目計算文と挿入文とを含む出力処理文を生  
 成し、処理種別が更新の場合には処理条件文と更新文と  
 を含む出力処理文を生成し、処理種別が削除の場合には  
 処理条件文と削除文とを含む出力処理文を生成する出力  
 処理文生成手段、  
 前記データ処理プログラムのうち、仕様の如何に関わら  
 ず固定の部分のプログラムと仕様によって変化する部分  
 のプログラム断片の挿入位置とを含むプログラムスケル  
 トンの該当挿入位置に、前記プログラムスケルトンカス  
 タマイズ情報取得手段で取得したプログラム名とプログ  
 ラムカスタマイズ情報と、前記カーソル定義文生成手段  
 で生成したカーソル定義文と、前記作業変数定義文生成  
 手段で生成した作業変数定義文と、前記出力処理文生成  
 手段で生成した出力処理文とを挿入することによってプ  
 ログラムを完成させ、出力するプログラム自動合成手  
 段、  
 として機能させるプログラムを記録したコンピュータ可  
 読記録媒体。

#### 【発明の詳細な説明】

#### 【0001】

【発明の属する技術分野】 本発明はプログラムの自動生  
 成に関し、特に、情報処理システム上に構築されたデー  
 タベースの表のデータを参照してデータ処理を行い、処  
 理結果を再度データベースの表に格納したり、処理結果  
 に基づいてデータベースの表のデータを更新したり、削  
 除したりするデータ処理プログラムを自動的に生成する  
 技術に関する。

#### 【0002】

【従来の技術】 人間にとって理解し易い仕様記述から、  
 そこに記述されたユーザの要求を満足するプログラムを  
 自動的に生成するプログラム自動生成技術については、  
 従来より各種のプログラムを対象に研究、開発が進めら  
 れている。

【0003】 例えば特開平 1-147621 号公報（以  
 下、文献 1 と称す）では、バッチ処理業務の入出力とな  
 る、ファイル、帳票等の各入出力データに対し、（1）  
 該ファイル・帳票が含む各データ項目の桁数、型などの



属性を定義したデータレイアウト仕様情報、(2) 該出力ファイル・帳票が含む各データの値を得るための計算式仕様情報を入力する手段と、入力された該仕様情報より、対応するバッチ処理業務プログラムを生成する手段と有する情報処理システムにおいて、仕様情報の入力手段に、ファイル・帳票等のデータ入出力順序、計算順序等の実行順序情報を入力するステップを有し、プログラムを生成する手段に、該実行順序情報から、生成すべきプログラムのプログラム構造情報を生成するステップを有するプログラム自動生成方法が提案されている。

【0004】前記文献1に示された従来技術(以下、第1の従来技術と称す)の構成を図41に示す。図41を参照すると、第1の従来技術では、仕様入力部から入力された仕様のうち、入出力関係仕様とレイアウト仕様とからデータ定義部を生成し、入出力関係仕様と生成式仕様とからデータ編集部を生成し、入出力関係仕様とデシジョンテーブル仕様とからプログラム構造を生成し、プログラム結合部でデータ定義部とデータ編集部とプログラム構造とを結合し、プログラムを生成している。

【0005】また、特開平3-182932号公報(以下、文献2と称す)では、COBOL言語のプログラムにSQL言語のプログラムを埋め込んで記述する際の一般的な問題点として、COBOLの言語とSQL言語の仕様が異なり、プログラム作成に考慮すべき点が多く、ユーザの誤りを誘発したり、記述量の増大を招き生産性を低下させる点を解決し、誤りの防止と生産性向上を目的としたプログラム自動生成技術が提案されている。

【0006】前記文献2に示された従来技術(以下、第2の従来技術と称す)の構成を図42に示す。図42を参照すると、第2の従来技術では、処理仕様を入力し、データ部解析と手続き部解析とを行ない、定義体を入力して、SQL宣言節とSQL連絡領域展開文とを自動生成している。

【0007】

【発明が解決しようとする課題】このように各種のプログラムをターゲットにしてその自動生成技術が提案されているが、情報処理システム上に構築されたデータベースの表のデータをカーソル定義文を使用して参照、更新、削除するデータ処理プログラムの自動生成技術を記載した文献は見当たらない。即ち、第1の従来技術は、通常のファイルや帳票を扱うプログラムの自動生成であってデータベースの利用は想定されておらず、また第2の従来技術では、カーソル定義文の自動生成は行われていない。

【0008】一般に、通常のファイルや帳票を扱うプログラムでは、入力となるファイルからデータを取得し、計算処理を行い、新規ファイルにその結果を格納する。このため第1の従来技術でも、出力処理として新規ファイル・帳票の作成を前提としており、既存ファイルのデータの削除や変更は行っていない。しかし、データベ

スシステムの場合は、既存表のレコード単位の更新や削除を安全に実施できるという特長があるため、データベースを利用したデータ処理プログラムでは、どの表のどの部分を処理するかを示すカーソル定義文に従って、入力となる表からデータを取得し、計算処理を行い、その結果に基づいて、新規表を作成したり、既存表を更新したり、削除したりすることが、広く行われている。

【0009】このようなデータ処理プログラムを、従前と同様に、詳細設計書を参照しながら人手で作成する場合にあつては、工数が多く掛かり、人手によるミスが防止しきれず、詳細設計書とプログラムの一貫性が保証できないなどの問題点が生じる。また、大規模な業務用プログラムを開発する場合には、データベースやネットワークへのアクセス障害、システムの障害が発生した場合の処理等を業務要件と共に実装する必要があるが、業務用プログラムの開発経験が浅いプログラマには困難で、開発されたプログラムの品質の劣化や開発の生産性の低下等の問題点が生じる。このため、この種のデータ処理プログラムの自動生成技術は重要な課題となっている。

【0010】本発明の目的は、データベースの表のデータを参照、更新、削除するデータ処理プログラムを自動的に生成し得るようにすることにある。

【0011】また本発明の別の目的は、業務要求の仕様に記述すべき情報の種類を削減することにより、利用者の仕様記述の手間を削減し得るようにすることにある。

【0012】また本発明の他の目的は、同じ業務要求の仕様から様々なデータ処理プログラムを自動生成することができるようすることにある。

【0013】

【課題を解決するための手段】データベースの表のデータを参照、更新、削除するデータ処理プログラムを自動的に生成する本発明のデータ処理プログラム自動生成システムは、仕様から取得した情報に基づいて、カーソル定義文、作業変数定義文、出力処理文を生成する手段と、前記データ処理プログラムのうち、仕様の如何に関わらず固定の部分のプログラムと仕様によって変化する部分のプログラム断片の挿入位置とを含むプログラムスケルトンの該当箇所に、前記生成されたカーソル定義文、作業変数定義文、出力処理文、および前記仕様中のプログラムカスタマイズ情報を挿入して前記データ処理プログラムを完成させ、出力する手段とを含む。より具体的には、プログラム名と入力表項目定義とセレクト項目定義と出力表項目定義と出力表処理定義とセレクト条件定義とプログラムカスタマイズ情報とを含む仕様を入力する仕様入力手段と、前記仕様入力手段で入力された仕様からプログラムスケルトンをカスタマイズするための情報として、プログラム名とプログラムカスタマイズ情報を取り出すプログラムスケルトンカスタマイズ情報取得手段と、前記仕様入力手段で入力された仕様から入力表項目定義とセレクト項目定義とセレクト条件定義を

取り出すカーソル定義情報取得手段と、前記仕様入力手段で入力された仕様から出力表項目定義と出力表処理定義を取り出す出力項目処理情報取得手段と、前記カーソル定義情報取得手段で取得された情報からカーソル定義文を生成するカーソル定義文生成手段と、前記出力項目処理情報取得手段で取得された出力表項目定義のうち、処理種別が挿入である出力表に対応するものから作業変数定義文を生成する作業変数定義文生成手段と、前記出力項目処理情報取得手段で取得された出力表項目定義と出力表処理定義から、処理種別が挿入の場合は処理条件文と項目計算文と挿入文とを含む出力処理文を生成し、処理種別が更新の場合には処理条件文と更新文とを含む出力処理文を生成し、処理種別が削除の場合には処理条件文と削除文とを含む出力処理文を生成する出力処理文生成手段と、前記データ処理プログラムのうち、仕様の如何に関わらず固定の部分のプログラムと仕様によって変化する部分のプログラム断片の挿入位置とを含むプログラムスケルトンの該当挿入位置に、前記プログラムスケルトンカスタマイズ情報取得手段で取得したプログラム名とプログラムカスタマイズ情報と、前記カーソル定義文生成手段で生成したカーソル定義文と、前記作業変数定義文生成手段で生成した作業変数定義文と、前記出力処理文生成手段で生成した出力処理文とを挿入することによってプログラムを完成させ、出力するプログラム自動合成手段とを有する。

【0014】このように、利用者が作成した仕様から、カーソル定義文、作業変数定義文、出力処理文を生成し、予め用意されたプログラムスケルトンの該当挿入箇所にこれらを挿入することにより、データベースの表のデータをカーソル定義文を使って参照、更新、削除するデータ処理プログラムを自動的に生成することができる。また、仕様中のプログラムカスタマイズ情報もプログラムスケルトンの該当箇所に挿入できるので、利用者がプログラムカスタマイズ情報として指定した例えば初期処理や終了処理などを組込むことができ、プログラムのカスタマイズ化も容易に行える。更に、プログラムスケルトンが仕様の如何に関わらず固定的に有するプログラム部分の一部に、データベースアクセスでエラーが発生した場合の処理プログラムや、多人数で開発する場合に守らなければならないプログラム規約を遵守するプログラム等を事前に設定しておくことで、この部分を作成するユーザの負担を軽減でき、またユーザミスを防止できる。

【0015】また本発明のデータ処理プログラム自動生成システムは、仕様に記述すべき情報の種類を削減することにより、利用者の仕様記述の手間を削減し得るようするために、別の一構成として、カーソル定義文生成手段が、入力表項目定義とセレクト条件定義と出力表処理定義とからセレクト項目定義を作成し、この作成したセレクト項目定義とセレクト条件定義とからカーソル定

義文を生成する。これによって、利用者はセレクト項目定義を作成する手間を省くことができる。また、他の構成として、仕様入力手段が、処理で使用する全ての入力表名と出力表名を入力し、データベースの表定義情報から当該入力表と出力表の全ての項目名と型情報を検索して取得する。これによって、利用者は入力表項目定義および出力表項目定義を作成する手間を省くことができる。更に、入力仕様手段は、前記取得した入力表と出力表の全ての項目名と型情報を画面上に提示して仕様記述者による入力表項目定義および出力表項目定義以外の仕様情報の指定を可能にすることで、出力表項目定義、出力表処理定義、セレクト条件定義などの記述を容易ならしめている。

【0016】また本発明のデータ処理プログラム自動生成システムは、同じ業務要求の仕様から様々なデータ処理プログラムを自動生成することができるようにするために、仕様の如何に関わらず固定の部分のプログラムと仕様によって変化する部分のプログラム断片の挿入位置とを含むプログラムスケルトンを複数種類格納するプログラムスケルトンライブラリと、このプログラムスケルトンライブラリから利用者の指定した1つのプログラムスケルトンを選択し、前記プログラム自動合成手段に渡すプログラムスケルトン選択手段とを備えるようにしている。

【0017】

【発明の実施の形態】次に本発明の実施の形態の例について図面を参照して詳細に説明する。

【0018】〔第1の実施の形態〕図1は本発明のデータ処理プログラム自動生成システムの第1の実施の形態の構成例を示したブロック図、図17はその処理の概略を示すフローチャートである。まず、図1及び図17を参照して第1の実施の形態の概要を説明する。

【0019】図1において、業務要件を示す仕様1は、開発するプログラムが満足すべき仕様として、プログラム名と入力表項目定義とセレクト項目定義と出力表項目定義と出力表処理定義とセレクト条件定義とプログラムカスタマイズ情報とを含んでいる。まず、図17の1701において、仕様入力手段101はこのような仕様1を入力する。次に、プログラムスケルトンカスタマイズ情報取得手段102は、入力された仕様のうち、プログラムスケルトンをカスタマイズするために必要な情報として、プログラム名とプログラムカスタマイズ情報を取り出し、プログラム自動合成手段109に渡す(1702)。また、カーソル定義情報取得手段103は、入力された仕様のうち、カーソル定義文を生成するのに必要な情報として、入力表項目定義とセレクト項目定義とセレクト条件定義を取り出し、カーソル定義文生成手段105に渡す(1703)。更に、出力項目処理情報取得手段104は、入力された仕様のうち、作業変数定義文と出力処理文を生成するのに必要な情報として、出力表項目定義と出力表処理定義とを



出力項目処理情報として取り出し、作業変数定義文生成手段106と出力処理文生成手段107に渡す(1704)。

【0020】次にカーソル定義文生成手段105は、カーソル定義情報取得手段103から受けとった情報を加工し、データベースの表の内、仕様で指定された処理において参照される表と項目の定義を行なう、カーソル定義文を生成し、プログラム自動合成手段109に渡す(1705)。また、作業変数定義文生成手段106は、出力項目処理情報取得手段104から受けとった情報を加工し、生成プログラム中の、処理種別が挿入である全ての出力表の各項目の計算結果を保存するための変数定義文を生成し、プログラム自動合成手段109に渡す(1706)。更に、出力処理文生成手段107は、出力項目処理情報取得手段104から受けとった情報を加工し、生成プログラム中の全ての出力表のうち、処理種別が挿入の出力表に対しては処理条件文と項目計算文と挿入文とからなる出力処理文を生成し、処理種別が更新の出力表に対しては処理条件と更新文とからなる出力処理文を生成し、処理種別が削除の出力表に対しては処理条件文と項目計算文と削除文とからなる出力処理文を生成し、プログラム自動合成手段109に渡す(1707)。

【0021】また図1において、プログラムスケルトン108は、生成プログラムのうち、仕様としてどのような業務要件が与えられても変化しない部分を、プログラムの骨組みとして用意したものである。このプログラムの骨組みにはプログラム名、カーソル定義文、作業変数定義文、出力処理文の部分と、さらに必要に応じて1つ以上のプログラムの断片の挿入位置が埋め込まれており、これらの挿入位置に、仕様で与えられた業務要件から生成されたプログラムの断片を挿入することにより、完成したプログラムを得ることを目的としたものである。

【0022】プログラム自動合成手段109は、プログラムスケルトン108の挿入位置の内、プログラム名に対応する挿入位置にプログラムスケルトンカスタマイズ情報取得手段102で取得されたプログラム名を挿入し、カーソル定義文に対応する挿入位置にカーソル定義文生成手段105で生成されたカーソル定義文を挿入し、作業変数定義文に対応する挿入位置に作業変数定義文生成手段106で生成された作業変数定義文を挿入し、出力処理文に対応する挿入位置に出力処理文生成手段107で生成された出力処理文を挿入し、その他の挿入位置にプログラムスケルトンカスタマイズ情報取得手段102で取得した対応するプログラム文の断片を挿入することによりプログラムを自動合成する(1708)。

【0023】次に本発明の第1の実施の形態の詳細を図1から図5を参照して説明する。

【0024】図2(a)は図1の仕様1の構成例を示した図である。201は仕様全体を表しており、図1における仕様1に対応する。図2(a)に示したように、仕様201は、仕様から生成したプログラムが持つべき名前

あるプログラム名202と、入力表の名前と各項目の名前と型の情報を含む入力表項目定義203と、入力表項目のうち、本仕様で指定する処理で実際に使用するため、セレクト文内に定義する項目の名前を含むセレクト項目定義204と、出力表の名前と各項目の名前と型の情報を含む出力表項目定義205と、各出力表の処理を行う条件を示す処理条件と、挿入、更新、削除の別を示す処理種別と、処理種別が更新と削除の場合に更新／削除するための条件を示す更新／削除条件と、処理種別が挿入と更新の場合に各項目の値を計算する処理式とを含む出力表処理定義206と、入力表からデータを取得する際の条件を含むセレクト条件定義207と、プログラムスケルトンをカスタマイズするために必要なその他の情報を含むプログラムカスタマイズ情報208とから構成される。

【0025】出力表処理定義206に必要な、処理種別毎の情報を図2(b)に一覧表で示す。図2(b)を参照すると、処理種別が挿入の場合には、処理式は必須であり、処理条件は任意であり、更新／削除条件は不要である。処理種別が更新の場合には、処理式と更新条件は必須であり、処理条件は任意である。処理種別が削除の場合には、削除条件は必須であり、処理条件は任意であり、処理式は不要である。なお、処理条件が指定されなかった場合には、常に処理条件が成立するものとして、その処理が行われる。図1の仕様入力手段101は、図2(a)のような構成を持つ仕様を入力し、プログラムスケルトンカスタマイズ情報取得手段102と、カーソル定義情報取得手段103と出力項目処理情報取得手段104に渡す。

【0026】プログラムスケルトンカスタマイズ情報取得手段102は、仕様入力手段101から渡された仕様情報から、図2のプログラム名202とプログラムカスタマイズ情報208に対応する部分を取り出し、プログラム自動合成手段109に渡す。カーソル定義情報取得手段103は、仕様入力手段101から渡された仕様情報から、図2の入力表項目定義203とセレクト項目定義204とセレクト条件定義207に対応する部分を取り出し、カーソル定義文生成手段105に渡す。出力項目処理情報取得手段104は、仕様入力手段101から渡された仕様情報から、図2の出力表項目定義205と出力表処理定義206に対応する部分を取り出し、作業変数定義文生成手段106と出力処理文生成手段107に渡す。

【0027】次にカーソル定義文生成手段105の動作について、図3を用いて説明する。図3は、カーソル定義文生成手段105の動作を表す流れ図である。図1のカーソル定義文生成手段105は、図3の301でカーソル定義情報取得手段103から入力表項目定義情報とセレクト項目定義情報とセレクト条件定義情報をカーソル定義情報として受けとる。302では、入力表項目定義情報のうち、セレクト項目定義情報内に指定された項目名だけを抜きだし、生成するプログラミング言語の構文に従った形式でセレクト項目列を作成する。303では、入力表項

目定義情報内に指定された入力表のうち、セレクト項目定義情報で使用されている入力表の名前だけを抜きだし、生成するプログラミング言語の構文に従った形式でセレクト表列を作成する。304では、セレクト条件定義情報を、生成するプログラム言語の構文に従った形式に加工し、セレクト条件文を作成する。305では、302、303、304でそれぞれ作成した、セレクト項目列、セレクト表列、セレクト条件をカーソル定義スケルトンの対応する挿入位置に挿入し、カーソル定義文を作成する。306でこのカーソル定義文を出力し、図1のプログラム自動合成手段109に渡す。

【0028】図7にカーソル定義スケルトンの構成例を示す。図7を参照すると、カーソル定義文スケルトン701は、特定のプログラミング言語に依存したスケルトンで、セレクト項目列挿入位置702、入力表列挿入位置703、セレクト条件挿入位置704を持って構成されている。これらの空白部分に、図1のカーソル定義文生成手段で作成した、それぞれ、セレクト項目列、入力表列、セレクト条件が挿入され、カーソル定義文となる。

【0029】次に作業変数定義文生成手段106の動作について、図4を用いて説明する。図4は、作業変数定義文生成手段106の動作を表す流れ図である。図1の作業変数定義文生成手段106は、図4の401で出力項目処理情報取得手段104から出力表項目定義情報を受け取る。出力表は一般に1つ以上指定可能だが、処理種別が更新や削除である出力表については、作業変数定義が不要なため、処理種別が挿入である出力表について、402が繰り返される。402の中では、ステップ403から405が行なわれる。403において、出力表のレコード型を定義する型定義文を作成する。404において、作業領域変数を403で定義された型で定義する文を作成する。405では、403で作成した型定義文と404で作成した変数定義文を作業変数定義スケルトンの対応する挿入位置に挿入する。406ステップでは、402の繰り返しによって完成した作業変数定義文列を出力し、図1のプログラム自動生成手段109に渡す。

【0030】図8に作業変数定義スケルトンの構成例を示す。図8を参照すると、作業変数定義スケルトン801は、型定義文挿入位置802と変数定義文挿入位置803を持って構成されている。図4の405では、これらの挿入位置にそれぞれ、型定義文、変数定義文を挿入し、作業変数定義文を完成させる。作業変数定義文が複数ある場合には、完成した複数の作業変数定義文をまとめて作業変数定義文列とする。

【0031】次に出力処理文生成手段107の動作について、図5を用いて説明する。図5は、出力処理文生成手段107の動作を表す流れ図である。図1の出力処理文生成手段107は、図5の501で出力項目処理情報取得手段104から出力表項目定義情報と出力表処理定義情報を受け取る。出力表は一般に1つ以上指定可能なので、全て

の出力表について、502が繰り返される。503において、501で受け取った出力表処理定義情報のうち、処理条件情報から処理条件文が作成される。504では、501で受け取った出力表処理定義情報のうち、処理種別情報が検査され、その値によって以後の処理が分岐する。処理種別情報が挿入の場合には、ステップ505から507が、処理種別情報が更新の場合には、ステップ508から509が、処理種別情報が削除の場合には、ステップ510から511が、実行される。

【0032】505では、出力表処理定義情報のうち、各項目の処理式を加工し、項目計算文を作成する。506では、出力表項目定義情報を加工し、挿入文を作成する。507では、503で作成した処理条件文と、505で作成した項目計算文と、506で作成した挿入文を出力処理スケルトンの対応する挿入位置に挿入し、出力処理文を完成する。

【0033】508では、出力表処理定義情報のうちの各項目の処理式と、出力表項目定義情報とを加工し、更新文を作成する。509では、503で作成した処理条件文と、508で作成した更新文と出力処理文スケルトンの対応する挿入位置に挿入し、出力処理文を完成する。

【0034】510では、出力表項目定義情報を加工し、削除文を作成する。511では、503で作成した処理条件文と、510で作成した削除文を出力処理文スケルトンの対応する挿入位置に挿入し、出力処理文を完成する。なお、出力種別が更新と削除の場合には、出力処理文スケルトンの項目計算文挿入位置には何も挿入しない。

【0035】512では、502の繰り返しによって完成した全ての出力処理文列を出力し、図1のプログラム自動生成手段109に渡す。

【0036】図9(a)に出力処理文スケルトンの構成例を示す。図9(a)を参照すると、出力処理文スケルトン901は、処理条件文挿入位置902と項目計算文挿入位置903と出力文挿入位置904を持って構成されている。図5の507、509、511では、これらの挿入位置にそれぞれ、処理条件文、項目計算文、挿入文／更新文／削除文を挿入し、出力処理文を完成させる。図9(a)を参照すると、処理条件文挿入位置902に挿入された処理条件がif文の条件節になっており、この条件が満足する場合だけ、項目計算文挿入位置903に挿入された出力項目の計算処理と出力文挿入位置904に挿入された出力表への出力処理が実行されることがわかる。

【0037】図9(b)から図9(d)に図9(a)の出力文挿入位置904に挿入される、挿入文、更新文、削除文のPL／SQL言語におけるテンプレートの例をそれぞれ示す。図9(b)を参照すると、処理種別が挿入の場合、出力表項目定義情報から、出力表の名前を取り出して<<出力表名>>に代入し、項目の名前を取り出して、”,”でつないで項目名のリストを作成し、<<項目名リスト>>に代入し、項目の名前を取り出し、これをも



とに、ある決まった方法で変数名を作成し、” , ” でつないで変数名のリストを作成、<<変数名リスト>>に代入することで、挿入文を作成する。

【0038】図9(c)を参照すると、処理種別が更新の場合、出力表項目定義情報から、出力表の名前を取り出して<<出力表名>>に代入し、出力表処理定義情報から処理式を取り出して、” , ” でつないで更新文リストを作成し、<<更新文リスト>>に代入し、出力表処理定義情報から更新条件を取り出して<<更新条件>>に代入することで、更新文を作成する。

【0039】図9(d)を参照すると、処理種別が削除の場合、出力表項目定義情報から、出力表の名前を取り出して<<出力表名>>に代入し、出力表処理定義情報から削除条件を取り出して<<削除条件>>に代入することで、削除文を作成する。

【0040】次に図1のプログラム自動合成手段109の動作について、図10と図6を用いて説明する。図10はプログラムスケルトンの構成例を示した図である。プログラムスケルトンは、ある特定のプログラミング言語で記述されたプログラムを仕様により変化する部分と、仕様により変化しない部分に分類し、仕様により変化しない部分は、プログラム文として持ち、仕様により変化する部分は、仕様から作成されたプログラムの断片の挿入位置を持つことで構成される。図10の例では、このプログラムの断片の挿入位置として、プログラム名挿入位置1002と、カーソル定義文挿入位置1003と、作業変数定義文列挿入位置1004と、” 初期処理” 挿入位置1005と、出力処理文列挿入位置1006と、” 終了処理” 挿入位置1007を持っている。

【0041】図6は、プログラム自動合成手段109の動作を表す流れ図である。プログラム自動合成手段109は、プログラムスケルトン108のうち、プログラムの断片の挿入位置に、仕様から作成したプログラムの断片を挿入することによりプログラムを完成させるものである。図6を参照すると、まず601ステップで、図1のプログラムスケルトン108を取り出す。602では、プログラムカスタマイズ情報とカーソル定義文と作業変数定義文列と出力処理文列を、それぞれ、図1のプログラムスケルトンカスタマイズ情報取得手段102、カーソル定義文生成手段105、作業変数定義文生成手段106、出力処理文生成手段107から受けとる。ステップ603では、602で受けとったカーソル定義文をプログラムスケルトンのカーソル定義文挿入位置に挿入する。604では、602で受けとった作業変数定義文列をプログラムスケルトンの作業変数定義文列挿入位置に挿入する。605では、602で受けとった出力処理文列をプログラムスケルトンの出力処理文列挿入位置に挿入する。606では、602で受けとったプログラム名をプログラムスケルトンのプログラム名挿入位置に挿入し、プログラムスケルトンカスタマイズ情報を、名前に対応が示された挿入位置に挿入す

る。図10のプログラムスケルトンでは、” 初期処理” と” 終了処理” がこれに当たる。ただし、仕様の記述とプログラムスケルトンが対応する限り、いくつかの挿入位置を設けることも可能である。607では、全てのプログラム断片が挿入され、完成したプログラムを出力する。

【0042】以上により、仕様からプログラムを自動的に作成し、出力できる。

【0043】なお、本説明では、カーソル定義スケルトン、作業変数定義スケルトン、出力処理文スケルトン、プログラムスケルトンは、PL/SQL言語に従って記述し、説明しているが、他のプログラミング言語のプログラムを自動生成する場合には、生成するプログラミング言語に従ったスケルトンを用意することで、他のプログラミング言語のプログラムを生成することが可能である。

【0044】また、図3、図4、図5、図6の処理流れ図において、横に並列して記述されている部分は、その間の処理順序がどのような順序であっても良いことを表している。例えば、図3において、302と303と304は、この順序で処理しても良いし、304、303、302の順序で処理しても良いし、あるいは、これらを合わせて並列に処理しても良いことを表している。

【0045】このように、本実施の形態では、プログラムの開発者は、仕様として、プログラム名と入力表項目定義とセレクト項目定義と出力表項目定義と出力表項目計算処理定義とセレクト条件定義とプログラムカスタマイズ情報を指定するだけで、データ処理プログラムを自動生成させることができる。したがって、従来、プログラミング工程において混入していたミス無くすことができ、またプログラム開発の生産性も大幅に向上させることができる。

【0046】〔第2の実施の形態〕次に本発明の第2の実施の形態について、第1の実施の形態との相違点を中心に説明する。

【0047】第1の実施の形態において、図2の仕様201は、プログラム名202と入力表項目定義203とセレクト項目定義204と出力表項目定義205と出力表項目計算処理206とセレクト条件207とプログラムカスタマイズ情報208から構成されていた。第2の実施の形態では、このうち、セレクト項目定義204を持たずに構成される。すなわち、図11に示したように、プログラム名1102と入力表項目定義1103と出力表項目定義1104と出力表処理定義1105とセレクト条件定義1106とプログラムカスタマイズ情報1107から構成されている。

【0048】第2の実施の形態では、図1の仕様入力手段101は、図11のような構成を持つ仕様を入力し、プログラムスケルトンカスタマイズ情報取得手段102とカーソル定義情報取得手段103と出力項目処理情報取得手段104に渡す。第2の実施の形態における図1のカーソル定義情報取得手段103は、仕様入力手段101から渡さ



れた仕様情報から、入力表項目定義1103と出力表処理定義1105とセレクト条件定義1106に対応する部分を取り出し、カーソル定義文生成手段105 に渡す。

【0049】次に第2の実施の形態におけるカーソル定義文生成手段105 の詳細な処理を図12に示す。図12の1201では、カーソル定義情報取得手段103 から入力表項目定義情報とセレクト条件定義情報と出力表処理定義情報をカーソル定義情報として受けとる。1202では、入力表項目定義情報と出力表処理定義情報から、図13に示すようにして、セレクト項目定義情報を作成する。以後は、図3に示した第1の実施の形態におけるカーソル定義文生成手段105 の処理方式と同一である。

【0050】図13を参照すると、1301でまず、セレクト表列とセレクト項目列を空とする。1302では、全ての入力表<<表>>の全ての項目<<項目>>について1303から1306を繰り返す。1303では、<<表>>と<<項目>>の組が、つまり注目している入力表の項目が、出力表処理定義内の処理条件または処理式または更新／削除条件で使用されているかどうかをテストしている。使用されていない場合には次の繰り返し要素に移る。使用されている場合には、1304で<<表>>と<<項目>>の組をセレクト項目列に追加する。次いで1305で<<表>>がセレクト表列内にあるかどうかをテストしている。既にあった場合には、次の繰り返し要素に移る。無い場合には、1306で<<表>>をセレクト表列に追加する。全ての入力表の全ての項目について以上を繰り返した後、1307でセレクト表列とセレクト項目列を合わせてセレクト項目定義情報として出力する。

【0051】以上により、本発明の第1の実施の形態におけるセレクト項目定義情報を自動的に作成することができた。したがって、図12の1203から1207の処理において、第1の実施の形態における図3と同一の動作でカーソル定義文を作成することができる。

【0052】他の構成要素の動作は第1の実施の形態と同一である。

【0053】本実施の形態では、利用者は仕様内にセレクト項目情報を指定することなく、第1の実施の形態と同じプログラムを自動生成ことができるため、第1の実施の形態と比較して、より容易に仕様を作成することができる、という効果がある。

【0054】〔第3の実施の形態〕次に本発明の第3の実施の形態について第1の実施の形態との相違点を中心に説明する。

【0055】図14に本発明の第3の実施の形態における構成図を示す。図14を図1と比較すると、プログラムスケルトンを一つ含む図1のプログラムスケルトン108の代わりにプログラムスケルトンを1種類以上いくつでも保持できるプログラムスケルトンライブラリ1411を持ち、プログラムスケルトン特定情報取得手段1404とプログラムスケルトン選択手段1412が追加されている。

【0056】仕様1401は、図1の仕様1と比較して、プログラムスケルトン特定情報を合わせ持つて構成される。この仕様の構成を図15に示す。図15を参照すると、仕様1501は、プログラム名1502と入力表項目定義1503とセレクト項目定義1504と出力表項目定義1505と出力表処理定義1506とセレクト条件定義1507とプログラムスケルトン特定情報1508とプログラムカスタマイズ情報1509から構成されている。

【0057】図14の仕様入力手段1403は、図15のような構成を持つ仕様を入力し、プログラムスケルトン特定情報取得手段1404と、スケルトンカスタマイズ情報取得手段1405と、カーソル定義情報取得手段1406と、出力項目処理情報取得手段1407に渡す。図14のプログラムスケルトン特定情報取得手段1404は、仕様入力手段1403で入力された仕様からプログラムスケルトン特定情報を取り出し、プログラムスケルトン選択手段1412に渡す。プログラムスケルトン選択手段1412は、プログラムスケルトン特定情報を用いて特定されたプログラムスケルトンをプログラムスケルトンライブラリ1411から取り出し、このプログラムスケルトンをプログラム自動合成手段1413に渡す。プログラムスケルトン特定情報を用いてプログラムが特定できない場合には、仕様のエラーとして利用者にその旨通知する。プログラム自動合成手段1413は、プログラムスケルトン選択手段1412から受け取ったプログラムスケルトン中の、プログラム断片の挿入位置に、プログラムスケルトンカスタマイズ情報取得手段1405から受け取ったプログラム名とプログラムカスタマイズ情報と、カーソル定義文生成手段1408から受け取ったカーソル定義文と、作業変数定義文生成手段1409から受け取った作業変数定義文列と、出力処理文生成手段1410から受け取った出力処理文列とを、第1の実施の形態と同一の方式で挿入し、プログラムを完成して出力する。

【0058】他の構成要素の動作は第1の実施の形態と同一である。

【0059】本実施の形態では、様々な処理形態を実装したプログラムスケルトンを複数用意することにより、様々なデータ処理プログラムを自動生成することができる。例えば、複数のプログラミング言語によるプログラムや、異なるアルゴリズムに基づくプログラム、異なる異常処理方式に基づくプログラムを自動生成できる。より広範なプログラムを自動生成できるため、第1の実施の形態と比較して適用範囲が広がる、という効果がある。

【0060】〔第4の実施の形態〕次に本発明の第4の実施の形態について第1の実施の形態との相違点を中心に説明する。

【0061】本実施の形態では、図1の仕様入力手段101が仕様を入力する際にデータベース定義情報を参照して仕様を入力する点が第1の実施の形態と異なる。なお、データベース定義情報は、自動生成しようとするデ

ータ処理プログラムが処理の対象とするデータベースの定義情報であり、図 1 には図示しない記憶部に格納されており、仕様入力手段 101 によって参照可能になっている。図 16 に第 4 の実施の形態における図 1 の仕様入力手段 101 の処理の流れ図を示す。

【0062】図 16 を参照すると、図 1 の仕様入力手段 101 は、1601 と 1602 でまず、処理で使用する入力表と出力表の名前だけを取得する。次いで、1603 と 1604 とで、それぞれ、入力表と出力表の名前からデータベース定義情報を検索し、該当する入力表と出力表の項目定義情報を取得する。項目定義情報には、項目名、型、項目へのコメント等がある。1605 では、1603 と 1604 で取得した入力表と出力表の項目定義情報を画面上に提示する。1606 では、1605 で提示した入力表と出力表の項目定義情報を画面上で仕様入力者に利用させることにより、プログラム名、セレクト項目定義、出力表処理定義、セレクト条件定義、プログラムカスタマイズ情報を指定させ、取得する。入力表項目定義と出力表項目定義は、仕様入力手段 101 がデータベース定義情報から検索したものが使用されるため、利用者は作成する必要はない。なお、第 2

の実施の形態に適用する場合には、セレクト項目定義の作成は不要である。

【0063】なお、図 16 の 1605 と 1606 で入力表と出力表の項目定義情報を仕様入力者に提示し、これを利用させることによって、プログラム名、セレクト項目定義、出力表処理定義、セレクト条件定義、プログラムカスタマイズ情報を指定させ、取得する方法としては、以下の (1) と (2) の二種類がある。

【0064】(1) 入力表と出力表の名前と各項目の項目名と型をマトリクスの一つの軸に沿って並べ、セレクト条件定義とプログラムカスタマイズ情報と出力表の各項目の名前および型をマトリクスの他の軸に沿って並べ、仕様内のセレクト項目定義とセレクト条件定義とプログラムカスタマイズ情報と出力表処理定義の指定の際に、入力表の各項目の情報を参照する時、マトリクスの対応するセルに記号をつけることによって、入力項目の値を参照可能とする。

(2) 入力表と出力表の名前と各項目の項目名、型を一覧表の形式で画面上に表示し、仕様入力者がセレクト項目定義とセレクト条件定義とプログラムカスタマイズ情報と出力表処理定義を指定する際に、画面上の表示領域の表名や項目名の上をマウスでクリックすることにより、該当する項目名や表名を文字情報として指定の領域に展開することで、表名や項目名を再度入力することなく仕様を入力可能とする。

【0065】図 1 の仕様 1 において、図 2 の入力表項目定義 203 および出力表項目定義 205 については、データベースの表定義と同一の情報であり、データベースを作成する際に既に作成され、データベース内に格納されていることが普通である。本実施の形態では、処理の仕様

として、改めて入力表項目定義および出力表項目定義を利用者に指定させるのではなく、利用者には、入力表と出力表の名前だけを指定させ、入力表項目定義と出力表項目定義はデータベース内に既に格納されている情報から自動的に検索し、取得することにより、利用者の仕様記述のための手間を削減し、また、人手で指定することによるミスを削減することができる、という効果がある。

【0066】〔第 5 の実施の形態〕次に本発明の第 5 の実施の形態について第 1 の実施の形態との相違点を中心に説明する。

【0067】本実施の形態では、図 1 のプログラムスケルトン 108 は、デバッグ用トレース文をその挿入箇所が明確に識別できるように例えば特殊な記号で囲んで挿入されている。また、仕様 1 には、デバッグ用のプログラムを生成するのか、本番用のプログラムを生成するのかの別を示す生成プログラム識別情報が含まれている。この生成プログラム識別情報は、他の仕様情報と一緒に仕様入力手段 101 により入力され、プログラム自動合成手段 109 に渡される。プログラム自動合成手段 109 は、この生成プログラム識別情報に従って、プログラムスケルトン 108 中のデバッグ用トレース文を取り去るか、残すかを切り分ける。即ち、本番用のプログラムを生成する場合には、プログラムスケルトン 108 中の特殊な記号で囲まれたデバッグ用トレース文をその特殊な記号とともに取り去り、デバッグ用のプログラムを生成する場合には、プログラムスケルトン 108 中の特殊な記号で囲まれたデバッグ用トレース文をそのまま残す。このとき、特殊な記号は取り去っても良く、あっても支障ない場合には残しておいても良い。

【0068】本実施の形態によれば、仕様 1 中の生成プログラム識別情報によって、デバッグ用のプログラムの生成と本番用のプログラムの生成とを切り替えることができるという効果がある。

【0069】〔第 6 の実施の形態〕次に本発明の第 6 の実施の形態について図 18 を用いて説明する。図 18 を参照すると、本発明の第 6 の実施の形態は、CPU、主記憶、各種の入出力装置を含むコンピュータ 1800 と、記録媒体 1803 とから構成される。なお、仕様 1801 は図 1 の仕様 1 に相当し、生成プログラム 1802 は図 1 の生成プログラム 2 に相当する。

【0070】記録媒体 1803 は、磁気ディスク、半導体メモリ、CD-ROM その他の機械読み取り可能な記録媒体であり、データ処理プログラム自動生成プログラムが記録されている。記録媒体 1803 に記録されたデータ処理プログラム自動生成プログラムは、コンピュータ 1800 の立ち上げ時などにコンピュータ 1800 に読み込まれ、コンピュータ 1800 の動作を制御することにより、コンピュータ 1800 上に、仕様入力手段 1804 と、プログラムスケルトンカスタマイズ情報取得手段 1805 と、カーソル定義情報



取得手段1806と、出力項目処理情報取得手段1807と、カーソル定義文生成手段1808と、作業変数定義文生成手段1809と、出力処理生成手段1810と、プログラムスケルトン1811と、プログラム自動合成手段1812とを実現する。これらの機能部は図1に示した同一名称のものと同じ機能を果たす。

【0071】なお、本説明では、第1の実施の形態について説明したが、第2、3、4、5の実施の形態についても同様に、対応するデータ処理プログラム自動生成プログラムを記録媒体1803に記録しておくことができる。

#### 【0072】

【実施例】次に本発明のデータ処理プログラム自動生成システムの実施例について、図1と図19から図40を用いて詳細に説明する。本実施例は、本発明の第2の実施の形態における実施例である。この実施例では、図19に示す仕様から図40に示すPL/SQLというプログラム言語で記述されたプログラムを自動生成する具体的システムについてその動作を説明する。

【0073】図19は、データ処理プログラム自動生成システムへの入力仕様の例である。本発明のデータ処理プログラム自動生成システムは、本仕様で指定された処理を実行するプログラムを自動的に生成するものである。

【0074】図19の例では、1901では生成されるプログラムが"URIAGE\_KEISAN"というプログラム名を持つことを指定している。1903から1910までは、この処理で使用されるデータベースの表の名前、入力なのか出力なのかの別、各表の項目名、各項目の型、さらに出力となる表については各項目の値を得るための計算処理が指定されている。本処理では、URIAESUとSHOUHINの二つの表を入力とし、URIAGEDAKAという一つの表を出力する。表URIAGESUは、項目CODEとURIAGEを持ち、それぞれ、char(4)、number(6)の型を持つ。char(4)は4桁の文字列であり、number(6)は6桁の数字であることを表している。表SHOUHINについても同様に、項目CODEとNAMEとPRICEを持ち、それぞれ、char(4)、char(20)、number(8)の型を持つ。表URIAGEDAKAは、項目CODE、NAME、TOTALを持ち、項目CODEの値として、表URIAGESUの項目CODEの値を代入することが、項目NAMEの値として、表SHOUHINの項目NAMEの値を代入することが、それぞれ、計算処理欄の"URIAGESU.CODE"、"SHOUHIN.NAME"で示されている。表URIAGEDAKAの項目TOTALの値として、表URIAGESUの項目URIAGEの値と表SHOUHINの項目PRICEの値を掛け合わせたものを代入することが、"URIAGESU.URIAGE \* SHOUHIN.PRICE"で示されている。1911では、表URIAGESUと表SHOUHINのデータをデータベースから取得する際の条件である、セレクト条件が指定されている。ここでは、表URIAGESUの項目URIAGEの値が10より大きい場合であって、かつ、表URIAGESUの項目CODEの値と表SHOUHINの項目CODEの値が等しい場合にだけ、表URIA

GESUと表SHOUHINからデータを入力し、処理することが示されている。1912と1913には、初期処理、終了処理として、それぞれ、"Init\_module;"、"End\_module;"という関数を呼び出すことが指定されている。

【0075】図19において、1901が図11のプログラム名1102に対応し、1903から1907が図11の入力表項目定義1103に対応し、1908から1910の表名、項目名、型の情報が図11の出力表項目定義1104に対応し、1908から1910の計算処理の情報が図11の出力表処理定義1105に対応し、1911が図11のセレクト条件定義1106に対応し、1912と1913が図11のプログラムカスタマイズ情報1107に対応する。

【0076】この仕様で表された処理は、商品コード(CODE)と売上数(URIAGE)を項目に持つ売上数表(URIAGESU)と、商品コード(CODE)と商品名(NAME)と価格(PRICE)を項目にもつ商品マスター表(SHOUHIN)を、同一商品コードで突き合わせ、売上数(URIAGESU)が10より大きい商品について、商品コード(CODE)と商品名(NAME)と売上総額(TOTAL)を項目にもつ売上高表(URIAGEDAKA)を作成することである。この時、売上高表の商品コード項目の値は売上数表の商品コード項目より代入し、売上高表の商品名の値は対応する商品コードの値を持つ商品マスター表の商品名項目より代入し、売上高表の売上総額項目の値を、売上数表の売上数項目の値と商品マスター表の価格項目の値の積を代入することで得ることを意味している。図20に図19に示した仕様の模式図を示す。ここでは、売上数表2001内のCODE項目の値、例えば2レコード目の0005と同一の値をCODE項目に持つ商品マスター表2002内のレコードは5レコード目のレコードであるため、これらのレコードが対応することになる。ここで、商品マスター2002より、商品コード0005の商品名がクッキーであり、価格が400円であることがわかる。これにより、出力表である売上高表2003に、商品コードとして0005、商品名としてクッキー、売上総額として8,000円(20個 \* 400円)を値に持つレコードを作成することを表している。

【0077】本実施例における、図1の仕様入力手段101の詳細な動作について図21を用いて説明する。図21は、仕様入力手段101の動作手順を示している。

【0078】図21を参照すると、2101:で、内部変数<<内部情報>>を空にし、2102:と2103:とでそれぞれ、"プログラム名:<<プログラム名>>"と"表:"という文字列を作成し、<<内部情報>>に追加している。ここで、ダブルクォートは文字列を表すが、文字列内の<<>>で囲われた項目、ここでは、<<プログラム名>>は、実際のプログラム名の値が文字列中のこの部分に入ることを表す。2104:では、2105:から2112:までの処理を、仕様内の各表について繰り返すことを示している。ここで、各繰り返しにおいて処理している表を<<表>>で表している。



【0079】2105: では、"( <<表>>, <<入出力区分>>, [" という文字列を作成し、<<内部情報>>に追加している。ここで、<<入出力区分>>は、" 入力" かまたは" 出力" であり、仕様の表の入出力区分の項目から得た値を指す。2106: では、<<入力区分>>が" 入力" か" 出力" かがテストされ、" 入力" だった場合には、2107: から2108: が、" 出力" だった場合には、2109: から2110: が実行される。2107: では、<<表>>の各項目について2108: が繰り返されることを示している。ここで、各項目の値を<<項目>>とする。2108: では、文字列"( <<項目>>, <<型>>)," を作成し、<<内部情報>>に追加している。2109: では、<<表>>の各項目について2110: が繰り返されることを示している。ここで、各項目の値を<<項目>>とする。2110: では、文字列"( <<項目>>, <<型>>, <<計算処理>>)," を作成し、<<内部情報>>に追加している。2111: は2106: から始まったIF文の最後であることを表している。2112: では、2108: または2110: の繰り返し処理で最後に付加された"," を削除し、文字列"])" を追加している。

【0080】2113: では、文字列" 処理条件: <<処理条件>>" を作成し、<<内部情報>>に追加することを示している。ここで、<<処理条件>>は、仕様のセレクト条件部分に指定された文字列を指す。2114: では、仕様に指定されたその他の属性について、2115: を繰り返すことを示している。2115: では、文字列、"<< 属性名>> : << 属性値>>" を作成し、<<内部情報>>に追加している。ここで、<<属性名>>、<<属性値>>は、それぞれ、仕様内の属性の名前とその値を示している。図2の仕様例では、" 初期処理"、" 終了処理" が属性名で" Init\_module;"、" End\_module;" が属性値である。

【0081】以上で、仕様内の情報が全て<<内部情報>>に取り込まれたことになる。図22に図19で示された仕様例を<<内部情報>>に取り込んだ例を示す。仕様入力手段101 は、この<<内部情報>>を、図1のプログラムスケルトンカスタマイズ情報取得手段102、カーソル定義情報取得手段103、出力項目処理情報取得手段104 に渡す。

【0082】プログラムスケルトンカスタマイズ情報取得手段102 は、仕様入力手段101 から渡された図22に示すような<<内部情報>>から、プログラム名とプログラムカスタマイズ情報を、図23に示す形式で取り出し、プログラム自動合成手段109に渡すものである。図23で、\* は"( <<属性名>> . << 属性値>>)" が属性名の数だけ繰り返されることを示している。なお、値はダブルクォートで囲われ、文字列として扱われる。プログラムスケルトンカスタマイズ情報取得手段102 では、図22に示すような<<内部情報>>が渡されると、2201: と2213: と2214: とからそれぞれ属性名と属性値を取り出し、図23の形式で<<プログラムスケルトンカスタマイズ情報>>を作成し、プログラム自動合成手段109 に渡す。図2

4はこの方式によって、図22の例から取得された<<プログラムスケルトンカスタマイズ情報>>の例を示したものである。

【0083】カーソル定義情報取得手段103 は、仕様入力手段101 から渡された<<内部情報>>から、入出力区分が入力である全ての表名と項目名の情報と、セレクト条件と、入出力区分が出力である全ての表の項目の計算処理の内容を取り出して<<カーソル定義情報>>を図25の形式で作成し、カーソル定義文生成手段105 に渡すものである。図25で、\* は直前の要素が必要個数分だけ繰り返されることを示している。図26は図25の形式に沿って図22の<<内部情報>>から取得された<<カーソル定義情報>>の例を示したものである。図1のカーソル定義情報取得手段103 では、図22の<<内部情報>>が渡されると、2202: から2206: の情報を図25の2501: の形式に変更し、<<カーソル定義情報>>内に、図26の2601: と2602: を作成する。次いで、図22の2211: から2212: の情報をそのまま抜きだし、<<カーソル定義情報>>内に追加する。最後に図22の2207: から2210: の情報から、計算処理に関する部分だけを抜きだし、図15の2503: の形式に変更し、<<カーソル定義情報>>内に、図26の2604: と2605: と2606: を作成する。

【0084】図1の出力項目処理情報取得手段104 は、仕様入力手段101 から渡された<<内部情報>>から、入出力区分が出力である全ての表名と項目名と型と計算処理の情報を取り出して、<<出力項目処理情報>>を図27の形式で作成し、作業変数定義文生成手段105 と出力処理文生成手段107 に渡すものである。図27で、\* は直前の要素が必要個数分だけ繰り返されることを示している。なお、型と計算処理の値はダブルクォートで囲われ、文字列として扱われる。

【0085】図28は図27の形式に沿って図22の<<内部情報>>から取得された<<出力項目処理情報>>の例を示したものである。出力項目処理情報取得手段104 では、図22の<<内部情報>>が渡されると、図22の2207: から2210: の情報を図27の2701: の形式に変更し、<<出力項目処理情報>>内に、図28の2801: から2807: を作成する。

【0086】次に図1のカーソル定義文生成手段105 の動作について図29と図30を用いて説明する。図30は生成しようとしているカーソル定義文のうち、仕様の如何に関わらず、固定的な骨組みを表すスケルトンである。3001: と3008: は "--" で始まっているため、コメント文である。3002: から3007: はデータベース操作のためのセレクト文のスケルトンであり、<<Select項目挿入位置>>と<<入力表名挿入位置>>と<<条件節挿入位置>>に仕様から得られた情報を加工して挿入することで完成したセレクト文になる。3009: では3002: から3007: で定義されたセレクト文に対応したカーソル用変数W\_0 が定義されている。本スケルトンの、3004: <<Select項目挿入位置>>と、3005: <<入力表名挿入位置>>と、3006:

<<条件節挿入位置>>にカーソル定義情報取得手段103 から受けとった図26の情報を基に作成したプログラム断片を挿入し、カーソル定義文を完成させて<<カーソル定義文>>とし、<<プログラム自動合成手段>>109 に渡すのがカーソル定義文生成手段105 である。

【0087】図30は実施の形態で説明した図7のカーソル定義スケルトンに対応し、特定の事例としてより詳細化したものである。図30の3004の<<セレクト項目列挿入位置>>、3005の<<入力表列挿入位置>>、3006の<<セレクト条件挿入位置>>は、それぞれ図7のセレクト項目列挿入位置702、入力表列挿入位置703、セレクト条件挿入位置704 に対応する。

【0088】次に図1のカーソル定義文生成手段105 の詳細な動作について図29を用いて説明する。2900: では、カーソル定義情報を図1のカーソル定義情報取得手段103 から受け取る。図19の仕様の例では、図26がカーソル定義情報である。2901: 2902: では、繰り返し処理で用いる変数<<入力表名列>><<Select項目列>>を空にし、初期化している。2903: では、カーソル定義情報内の入力表全てに対し、2904: から2912: の処理を繰り返すことを示している。繰り返しの中で、着目している入力表は<<入力表>>で表す。2904: では、カーソル定義情報内の<<入力表>>の全項目について、2905: から2912: の処理を繰り返すことを示している。繰り返しの中で、着目している項目は<<項目>>で表す。つまり、2905: から2912: の処理は、各<<入力表>>の各<<項目>>について繰り返し処理されることになる。

【0089】2905: では、<<入力表>><<項目>>の組合せがカーソル定義情報内の計算処理内で使用されているかどうかを検査されている。使用されている場合には、2906: で、"<< 入力表>>.<< 項目>> << 入力表>>.<< 項目>>,"という文字列を作成し、変数<<Select項目列>>に追加する。ここで、<<入力表>>、<<項目>>は、それぞれ着目している入力表、項目の値を示す。2907: ではさらに、<<入力表>>が<<入力表名列>>に存在するかどうかを検査されている。存在する場合には何もせず(2908:)、存在しない場合には2909: で、変数<<入力表名列>>の最後に変数<<入力表>>とコンマからなる文字列を追加する。

【0090】2913: と2914: では、それぞれ、変数、<<Select項目列>>と<<入力表名列>>の最後にあるコンマ","を削除する。これは、コンマはそれぞれの要素を区切るために必要なもので、最後には必要無いためである。2915: では、仕様内のセレクト条件として指定された内容を変数<<条件節>>にセットする。2916: から2918: では、それぞれ、変数、<<Select項目列>>、<<入力表名列>>、<<条件説>>の値を、図30に示したカーソル定義スケルトンの3004: <<セレクト項目列挿入位置>>、3005: <<入力表列挿入位置>>、3006: <<セレクト条件挿入位置>>に挿入し、カーソル定義文を完成させる。完成し

たカーソル定義文を図1のプログラム自動合成手段109 に渡す。

【0091】図29は第2の実施の形態で説明した図12のカーソル定義文生成手段の処理流れ図と対応し、特定の事例としてより詳細化したものである。図29において、2900は図12の1201に対応し、2901から2914までが図12の1202から1204に対応する。図29は特定の事例に関し、処理の最適化が施されているため、図13に示した図12の1202ステップのセレクト項目情報作成処理を実施すると同時に1203と1204を実施しているが、図29の2906が図12の1203に対応し、図29の2909が図12の1204に対応する。図29の2915が図12の1205に、図29の2916から2918が図12の1206に、図29の2919が図12の1207に対応する。

【0092】図31に、図29の方式で、図30のカーソル定義文スケルトンを図26の<<カーソル定義情報>>で完成したカーソル定義文を示す。図31と図30を比較すると、図30の<<セレクト項目列挿入位置>>、<<入力表列挿入位置>>、<<セレクト条件挿入位置>>に、それぞれ以下の文字列が挿入され、カーソル定義が完成していることがわかる。

<<セレクト項目列挿入位置>> :  
 URIAGESU.CODE URIAGESU\_CODE,  
 URIAGESU.URIAGE URIAGESU\_URIAGE,  
 SHOUHIN.NAME SHOUHIN\_NAME,  
 SHOUHIN.PRICE SHOUHIN\_PRICE  
 <<入力表列挿入位置>> :  
 URIAGESU, SHOUHIN  
 <<セレクト条件挿入位置>> :  
 URIAGESU.URIAGE > 10 and  
 URIAGESU.CODE = SHOUHIN.CODE

【0093】次に図1の作業変数定義文生成手段106 の動作について、図32と図33を用いて説明する。図33は生成しようとしている作業変数定義文のうち、仕様の如何に関わらず、固定的な骨組みを表すスケルトンである。3301: は"--"で始まっているため、コメント文である。3302: から3305: は、作業変数の型定義文のスケルトンである。3302に作業変数型名を3304: に出力項目定義情報を挿入することで完成した作業変数の型定義文になる。3306: は、出力用の作業変数の定義文の挿入位置である。本スケルトンの、  
 3302: <<作業変数型名挿入位置>>と、  
 3304: <<出力項目定義挿入位置>>と、  
 3306: <<変数定義文挿入位置>>と  
 に図1の出力項目処理情報取得手段104 から受けとった<<出力項目処理情報>>を基に作成したプログラム断片を挿入し、作業変数定義文を完成させてプログラム自動合成手段109 に渡すのが、作業変数定義文生成手段106 である。

【0094】図33は実施の形態で説明した図8の作業



変数定義スケルトンに対応し、特定の事例としてより詳細化したものである。図 3 3 の 3302 から 3305 までは図 8 の型定義文挿入位置 802 に対応する。図 3 3 では、図 8 の型定義文挿入位置 802 のうち、P L / S Q L 言語特有の、仕様によらない部分を固定的に持っているためにより詳細な<<作業変数型名挿入位置>>と<<出力項目定義挿入位置>>を持って構成されている。図 3 3 の 3306 <<変数定義文挿入位置>>は、図 8 の変数定義文挿入位置 803 と対応する。

【0095】図 3 2 を用いて本実施例における、図 1 の作業変数定義文生成手段 106 の詳細な動作を説明する。3200: では、図 2 8 の<<出力項目処理情報>>を図 1 の出力項目処理情報取得手段 104 から受け取る。3201: と 3202: は、繰り返し処理で用いる変数<<作業変数定義文列>>を空に、<<出力表番号>>を 1 にし、初期化している。3203: は、図 1 の出力項目処理情報取得手段 104 から得られた、<<出力項目処理情報>>内の出力表の個数回だけ、3204: から 3215: を繰り返すことを示している。なお、3204: から 3215: において、着目している出力表は<<出力表>>で表される。

【0096】3204: では、変数<<作業変数型名>>に"<<出力表>>\_<<出力表番号>>\_Type"の文字列がセットされる。ここで、<<出力表>>、<<出力表番号>>はそれぞれ、変数<<出力表>>、<<出力表番号>>の値に置き換えられる。3205: では、繰り返し処理で用いる変数<<出力項目定義>>を空にし、初期化している。

【0097】3206: では、<<出力項目処理情報>>内の各<<出力表>>に対する項目の個数回数だけ、3207: から 3208: を繰り返すことを示している。なお、3207: から 3208: において、着目している項目は<<項目>>で表される。3207: では、仕様内の<<出力表>>の<<項目>>に対応する型を<<型>>にセットし、3208: では、",<<項目>> <<型>>"の文字列を作成し、変数<<出力項目定義>>に追加する。この時、<<項目>>、<<型>>は、それぞれ、変数<<項目>>、<<型>>の値に置き換えられる。3209: と 3210: では、変数<<変数定義>>に"<<出力表>>\_<<出力表番号>> <<作業変数型名>>:"をセットする。<<作業変数型名>>は、3204 で、"<<出力表>>\_<<出力表番号>>Type;"にセットされているため、結局、変数<<変数定義>>には"<<出力表>>\_<<出力表番号>> <<出力表>>\_<<出力表番号>>Type;"がセットされる。

【0098】3211: から 3213: では、作業変数スケルトンの、それぞれ、<<作業変数型名挿入位置>>に<<作業変数型名>>を、<<出力項目定義挿入位置>>に<<出力項目定義>>を、<<変数定義文挿入位置>>に<<変数定義>>を挿入し、作業変数定義文を完成している。

【0099】3214: では、完成した作業変数定義文を<<作業変数定義文列>>に追加し、3215: では、出力表番号に 1 を加算している。3214: と 3215: は出力表が複数あ

った場合の、次の繰り返しのための準備である。全ての出力表に関する作業変数定義文が<<作業変数定義文列>>に追加された後、図 1 の作業変数定義文生成手段 106 は、<<作業変数定義文列>>をプログラム自動合成手段 109 に渡す。

【0100】図 3 4 に、図 3 2 の方式で、図 3 3 の作業変数スケルトンを図 2 8 の<<出力項目処理情報>>で完成した作業変数定義文を示す。図 3 4 と図 3 3 を比較すると、図 3 3 の<<作業変数型名挿入位置>>、<<出力項目定義挿入位置>>、<<変数定義文挿入位置>>に、それぞれ以下の文字列が挿入され、作業変数定義が完成していることがわかる。

【0101】<<作業変数型名挿入位置>>:

URIAGEDAKA\_1\_Type

<<出力項目定義挿入位置>>:

, CODE char(4)

, NAME char(13)

, TOTAL number(14)

<<変数定義挿入位置>>:

20 URIAGEDAKA\_1 URIAGEDAKA\_1\_Type

【0102】なお、図 2 8 の例では、出力表が一つしかないため、出力用作業変数定義文は一つしか生成されないが、出力表が複数あれば、出力表の数だけ型定義文と変数定義文が生成される。

【0103】図 3 2 は、実施の形態で説明した図 4 の作業変数定義文生成手段の処理流れ図と対応し、特定の事例としてより詳細化したものである。図 3 2 において、3200 は図 4 の 401 に対応し、3203 から 3215 までは図 4 の 402 の繰り返しに対応する。図 3 2 の 3204 は、図 4 の 403 と 404 の両方に対応する前準備である。図 3 2 の 3205 から 3208 までは、図 4 の 403 に対応し、型定義文を生成している部分である。図 3 2 の 3209 から 3210 までは図 4 の 404 に対応し、変数定義文を作成している部分である。図 3 2 の 3211 から 3213 までは図 4 の 405 に対応し、型定義文の部分である<<作業変数型名>>と<<出力項目定義>>と、変数定義文を作成している。図 3 2 の 3216 が図 4 の 406 に対応している。

【0104】次に図 1 の出力処理文生成手段 107 の動作について図 3 5 と図 3 6 を用いて説明する。図 3 6 は生成しようとしている出力文のうち、仕様の如何に関わらず、固定的な骨組みを表すスケルトンである。3601: と 3603: と 3611: は "--" で始まっているため、コメント文である。3602: には、出力表の各項目の値を計算する文の列が挿入されることを示している。3604: では、変数 SQLRESULT に 0 が代入されている。変数 SQLRESULT は、処理が正常に行なわれている間 0 の値を持つため、ここで 0 にセットしているのは変数の初期化に当たる。3605: から 3610: では、3602: に挿入された文によって計算された出力表の各項目の値を、データベース表に書き込む処理が、insert 文によって行なわれている。360



6: の<<出力表名挿入位置>>に指定された表の、3607: の<<出力項目名挿入位置>>に指定された項目の値を、3609: の<<作業変数名挿入位置>>に指定された変数の値とし、データベースに書き込む。3612: から3615: では、データベースへの書き込み処理でエラーが発生した場合に、データベースから渡されたエラーコードであるSQLCODE の値を変数SQLRESULT に代入している、固定的な処理である。

【0105】本スケルトンの、3602: <<項目計算文挿入位置>>と、3606: <<出力表名挿入位置>>と、3607: <<出力項目名挿入位置>>と、3609: <<作業変数名挿入位置>>とに出力項目処理情報取得手段104 から受けとった<<出力項目処理情報>>を基に作成したプログラム断片を挿入し、出力処理文を完成させて<<出力処理文>>とし、プログラム自動合成手段109 に渡すのが、出力処理文生成手段107 である。

【0106】図36は実施の形態で説明した図9の出力処理文スケルトンに対応し、特定の事例としてより詳細化したものである。図36の3602<<項目計算文挿入位置>>が図9の項目計算文挿入位置902 に対応する。図36の3606から3610までが図9の出力文挿入位置903: に対応する。図36では、図9の出力文挿入位置903 のうち、PL／SQL言語特有の、仕様によらない部分を固定的に持っているためにより詳細な<<出力表名挿入位置>>と<<出力項目名挿入位置>>と<<作業変数名挿入位置>>を持って構成されている。

【0107】次に出力処理文生成手段107 の詳細な動作について図35を用いて説明する。3500: では、図1の出力項目処理情報取得手段104 から出力項目処理情報を入力する。この事例では、出力項目処理情報は図28の形式となる。3501: と3502: は、繰り返し処理で用いる変数<<出力処理文列>>を空に、<<出力表番号>>を1にし、初期化している。3503: は、図1の出力項目処理情報取得手段104 から得られた出力項目処理情報内の、出力表の回数だけ、3504: から3521: を繰り返すことを示している。なお、3504: から3521: において、着目している出力表は<<出力表>>で表される。

【0108】3504: から3506: では、繰り返し処理で用いる変数<<計算文列>>、<<項目名列>>、<<変数名列>>をそれぞれ空にし、初期化している。3507: は、図1の出力項目処理情報取得手段104 から得られた出力項目処理情報内、<<出力表>>の項目の回数だけ、3508: から3513: を繰り返すことを示している。なお、3508: から3513: において、着目している項目は<<項目>>で表される。3508: と3509: では、出力項目処理情報内の<<出力表>>の<<項目>>に対応する計算処理欄の値の中に出現する、"." を "\_" に置換し、表名の前に"W\_0."を追加して、変数<<計算式>>にセットする。3510: と3511: で

は、文字列  
"<< 出力表>>\_<< 出力表番号>>.<< 項目>> := <<計算式>>";

を作成し、変数<<計算文列>>の最後に追加する。上記文字列内で、<<出力表>>、<<出力表番号>>、<<項目>>、<<計算式>>は、それぞれ変数<<出力表>>、<<出力表番号>>、<<項目>>、<<計算式>>の値に置き換えられる。3512: では、文字列"<<項目>>,"を、変数<<項目名列>>の最後に追加する。3513: では、文字列"<< 出力表>>\_<< 出力表番号>>.<< 項目>>,"を、変数<<変数名列>>の最後に追加する。上記文字列内で、<<出力表>>、<<出力表番号>>、<<項目>>はそれぞれ変数の値で置き換えられる。

【0109】3514: と3515: は、それぞれ<<項目名列>>と<<変数名列>>の最後の"," を削除する。これは、コンマはそれぞれの要素を区切るために必要なもので、最後には必要無いためである。3516: から3519: では、出力処理文スケルトンのそれぞれ、<<出力項目計算文挿入位置>>、<<出力表名挿入位置>>、<<出力項目名挿入位置>>、<<作業変数名挿入位置>>に変数、<<計算文列>>、<<出力表>>、<<項目名列>>、<<変数名列>>の値を挿入し、出力処理文を完成させる。3520: では、完成した出力処理文を<<出力処理文列>>の最後に追加し、3521: では、出力表番号に1を加算している。3520: と3521: は出力表が複数あった場合の、次の繰り返しのための準備である。全ての出力表に関する出力処理文が<<出力処理文列>>に追加された後、図1の作業変数定義文生成手段107 は、<<出力処理文列>>をプログラム自動合成手段109 に渡す。

【0110】図37に、図35の方式で、図36の出力処理文スケルトンを図28の<<出力項目処理情報>>で完成した出力処理文を示す。図37と図36を比較すると、図36の<<項目計算文挿入位置>>、<<出力表名挿入位置>>、<<出力項目名挿入位置>>、<<作業変数名挿入位置>>に、それぞれ以下の文字列が挿入され、出力処理が完成していることがわかる。

【0111】<<項目計算文挿入位置>> :  
URIAGEDAKA\_1.CODE := W\_0.URIAGESU\_CODE;  
URIAGEDAKA\_1.NAME := W\_0.SHOUHIN\_NAME;  
URIAGEDAKA\_1.TOTAL := W\_0.URIAGESU\_URIAGE \* W\_0.SHOUHIN\_PRICE;  
<<出力表名挿入位置>> :  
URIAGEDAKA  
<<出力項目名挿入位置>> :  
CODE,  
NAME,  
TOTAL  
<<作業変数名挿入位置>> :  
URIAGEDAKA\_1.CODE,  
URIAGEDAKA\_1.NAME,  
URIAGEDAKA\_1.TOTAL

【0112】図35は、実施の形態で説明した図5の出力処理文生成手段の処理流れ図と対応し、特定の事例としてより詳細化したものである。図35において、3500は図5の501に対応し、3503から3521までが図5の502の繰り返しに対応する。図35では、特定の事例における処理の最適化を行なっているため、図5における503と504は混在した形になっている。図35の3504と、3508から3511までが図5の503である、項目計算文の作成に対応する。図35の3505から3506までと、3512と3513と、3514から3515までが図5の504である、出力文の作成に対応する。図35の3516から3519までは、図5の505に対応している。図35の3522は図5の506に対応している。

【0113】次に本実施例における、図1のプログラム自動合成手段109とプログラムスケルトン108について説明する。図39に本実施例におけるプログラムスケルトンを示す。これは、本発明のデータ処理プログラム自動生成システムが出力するデータ処理プログラムのうち、仕様の如何に関わらず固定の部分のスケルトンである。図39には、<<プログラム名挿入位置>>、<<カーソル定義文挿入位置>>、<<作業変数定義文列挿入位置>>、<<"初期処理"挿入位置>>、<<出力処理文列挿入位置>>、<<"終了処理"挿入位置>>の6箇所に具体的なプログラムになっていない部分がある。この部分は、仕様が変わるとプログラムも変化する部分である。図1のプログラム自動合成手段109は、これら6箇所に、プログラムスケルトンカスタマイズ情報取得手段102、カーソル定義文生成手段105、作業変数定義文生成手段106、出力処理文生成手段107から渡されたそれぞれ、<<プログラムスケルトンカスタマイズ情報>>、<<カーソル定義文>>、<<作業変数定義文列>>、<<出力処理文列>>を挿入することにより、データ処理プログラムを完成させ、生成プログラム2を出力するものである。

【0114】図39は実施の形態で説明した図10のプログラムスケルトンに対応し、特定の事例としてより詳細化したものである。図39の3901:<<プログラム名挿入位置>>、3909:<<カーソル定義文挿入位置>>、3910:<<作業変数定義文列挿入位置>>、3914:<<"初期処理"挿入位置>>、3927:<<出力処理文挿入位置>>、3930:<<"終了処理"挿入位置>>が図10の、それぞれ、プログラム名挿入位置1002、カーソル定義文挿入位置1003、作業変数定義文列挿入位置1004、"初期処理"挿入位置1005、出力処理文挿入位置1006、"終了処理"挿入位置1007に対応している。

【0115】図38に本実施例に置ける図1のプログラム自動合成手段109の処理手順を示す。まず、3800:で図1のプログラムスケルトン108を入力する。3801で、<<プログラム名>>と<<プログラムカスタマイズ情報>>と<<カーソル定義文>>と<<作業変数定義文列>>と<<出力処理文列>>を受け取る。3802:から3804:で、図1のプログラムスケルトンカスタマイズ情報取得手段102:から渡された<<プログラムスケルトンカスタマイズ情報>>から、プログラム名、初期処理、終了処理に対応する文字列を、図39のそれぞれ、3901:<<プログラム名挿入位置>>、3914:<<初期処理挿入位置>>、3930:<<終了処理挿入位置>>に挿入する。3805:では、図1のカーソル定義文生成手段105から渡された<<カーソル定義文>>を図39の3909:<<カーソル定義文挿入位置>>に挿入する。3806:では、図1の作業変数定義文生成手段106から渡された<<作業変数定義文列>>を図39の3910:<<作業変数定義文列挿入位置>>に挿入する。3807:では、図1の出力処理文生成手段107から渡された<<出力処理文列>>を図39の3927:<<出力処理文挿入位置>>に挿入する。3808:では、以上により完成したデータ処理プログラムを図1における生成プログラム2として出力する。以上により、仕様からプログラムの自動生成が全て終了したことになる。

【0116】図40に、図19に示した仕様から出力された生成プログラムの例を示す。図40と図39を比較すると、図39の<<プログラム名挿入位置>>、<<カーソル定義文挿入位置>>、<<作業変数定義文挿入位置>>、<<初期処理文挿入位置>>、<<出力処理文挿入位置>>、<<終了処理文挿入位置>>にそれぞれ、以下の文が挿入され、プログラムが完成していることがわかる。

【0117】

```
URIAGE_KEISAN()
```

```
<<カーソル定義文挿入位置>>:
```

```
--カーソル定義
```

```
cursor CSR_0 is
```

```
select
```

```
URIAGESU.CODE URIAGESU_CODE,
```

```
URIAGESU.URIAGE URIAGESU_URIAGE,
```

```
SHOUHIN.NAME SHOUHIN_NAME,
```

```
SHOUHIN.PRICE SHOUHIN_PRICE
```

```
from URIAGESU, SHOUHIN
```

33

34

```

where URIAGESU.URIAGE>10
and URIAGESU.CODE=SHOUHIN.CODE
;

```

```

--カーソル用変数定義

```

```

W_O CSR_0%rowtype;

```

```

<<作業変数定義文挿入位置>> :

```

```

--出力用作業変数定義

```

```

type URIAGEDAKA_1_Type is record(
    W_ROWID rowid
    ,CODE char(4)
    ,NAME char(22)
    ,TOTAL number(14)
);

```

```

URIAGEDAKA_1 URIAGEDAKA_1_Type;

```

```

<<初期処理文挿入位置>> :

```

```

Init_module;

```

```

<<出力処理文挿入位置>> :

```

```

--各出力項目の計算

```

```

URIAGEDAKA_1.CODE := W_O.URIAGESU_CODE;

```

```

URIAGEDAKA_1.NAME := W_O.SHOUHIN_NAME;

```

```

URIAGEDAKA_1.TOTAL:=W_O.URIAGESU_URIAGE * W_O.SHOUHIN_PRICE

```

```

--出力表への出力処理

```

```

#SQLRESULT := 0;

```

```

begin

```

```

    insert into URIAGEDAKA(

```

```

        CODE

```

```

        ,NAME

```

```

        ,TOTAL_SALE

```

```

    ) values(

```

```

        URIAGEDAKA_1.CODE

```

```

        ,URIAGEDAKA_1.NAME

```

```

        ,URIAGEDAKA_1.TOTAL

```

```

    );

```

```

/*例外処理 */

```

```

--例外処理

```

```

exception

```

```

    when others then

```

```

        SQLRESULT := SQLCODE;

```

```

end;

```

```

<<終了処理文挿入位置>> :

```

```

End_module;

```

【0118】図38は、実施の形態で説明した図6のプログラム自動合成手段の処理流れ図と対応するものである。図38において、3800と3801は、それぞれ、図6の601と602に対応する。図38の3802から3804までが図6の606に対応し、図38の3805、3806、3807が、図6

のそれぞれ、603、604、605に対応する。図38の3808は、図6の607に対応する。

【0119】

【発明の効果】以上説明したように、本発明によれば、ソフトウェア開発者が、データベースの表のデータをカ



カーソル定義文を使って新規作成、更新、または削除を行うデータ処理プログラムを開発する際、処理の仕様を記述するだけで、そこからプログラムを自動的に生成し、利用することができる。

【0120】また、セレクト項目定義を自動生成する構成にあっては、その分だけソフトウェア開発者の仕様記述の手間が省け、また誤りを作り込む危険性を少なくすることができる。同様に、データベースの表定義情報から入力表項目定義、出力表項目定義を検索して取得する構成にあっては、その分だけソフトウェア開発者の仕様記述の手間が省け、また誤りを作り込む危険性を少なくすることができる。

【0121】また、複数種のプログラムスケルトンから任意のプログラムスケルトンを選択できる構成にあっては、同じ業務要求の仕様から、異なるプログラミング言語によるプログラム、異なるアルゴリズムによるプログラム、異なる異常系処理方式に基づくプログラム等、様々なデータ処理プログラムを自動生成することができる。

【0122】また、プログラムの全体が自動生成されるため、プログラム作成を手で行う必要がなくなり、プログラム開発の手間を大きく削減することができる。

【0123】また、プログラム開発において、人手の作業が大幅に削減されることから、人手作業によるミス混入の機会が大幅に減るため、プログラムへのミスの混入を大幅に削減することができる。

【0124】また、プログラムスケルトン内に、データベースやネットワークへのアクセス障害、システムの障害が発生した場合の処理等、経験豊富なプログラム開発者のノウハウが詰まった処理を埋め込んでおくことにより、業務要件だけを記述した仕様からこれらの複雑な処理を行なうプログラムを自動生成できるため、業務用プログラムの開発経験の浅いプログラム開発者でも、経験者と同等な高品質なプログラムを開発できる。

【0125】また、システムの保守をする際、プログラムではなく、仕様で保守を行うことができ、保守を大幅に容易化できることである。その理由は、仕様からプログラムを自動生成できると、仕様とプログラムが1対1に対応することになり、システムに修正が必要となった場合に、プログラムを直接修正するのではなく、仕様を修正し、修正された仕様からプログラムを自動生成することで、保守を行うことができるからである。

#### 【図面の簡単な説明】

【図1】本発明の第1の実施の形態の構成を示すブロック図である。

【図2】本発明の第1の実施の形態における仕様の構成と、仕様中の出力表処理定義における処理種別毎の必要情報を示す図である。

【図3】本発明の第1の実施の形態におけるカーソル定義文生成手段の処理を示す流れ図である。

【図4】本発明の第1の実施の形態における作業変数定義文生成手段の処理を示す流れ図である。

【図5】本発明の第1の実施の形態における出力処理文生成手段の処理を示す流れ図である。

【図6】本発明の第1の実施の形態におけるプログラム自動合成手段の処理を示す流れ図である。

【図7】本発明の第1の実施の形態におけるカーソル定義文生成手段で使用するカーソル定義文スケルトンの構成を示す図である。

10 【図8】本発明の第1の実施の形態における作業変数定義文生成手段で使用する作業変数定義文スケルトンの構成を示す図である。

【図9】本発明の第1の実施の形態における出力処理文生成手段で使用する出力処理文スケルトンの構成と、そのスケルトンのうちの挿入文、更新文、削除文の構成を示す図である。

【図10】本発明の第1の実施の形態におけるプログラムスケルトンの構成を示す図である。

20 【図11】本発明の第2の実施の形態における仕様の構成を示す図である。

【図12】本発明の第2の実施の形態におけるカーソル定義文生成手段の処理を示す流れ図である。

【図13】本発明の第2の実施の形態におけるセレクト項目定義情報を作成する処理を示す流れ図である。

【図14】本発明の第3の実施の形態の構成を示すブロック図である。

【図15】本発明の第3の実施の形態における仕様の構成を示す図である。

30 【図16】本発明の第4の実施の形態における仕様入力手段の処理を示す処理流れ図である。

【図17】本発明の第1の実施の形態の処理例を示す処理流れ図である。

【図18】本発明の第6の実施の形態の構成を示すブロック図である。

【図19】本発明の実施例の仕様の構成及び例を示す図である。

【図20】本発明の実施例の仕様の例の意味を説明するための図である。

【図21】本発明の実施例における仕様入力手段の処理を示す流れ図である。

【図22】本発明の実施例における、仕様を入力した後の内部情報の構成と例を示す図である。

【図23】本発明の実施例における、プログラムスケルトンカスタマイズ情報取得手段がプログラム自動合成手段に渡すプログラムスケルトンカスタマイズ情報の構成を示す図である。

50 【図24】本発明の実施例における、プログラムスケルトンカスタマイズ情報取得手段がプログラム自動合成手段に渡すプログラムスケルトンカスタマイズ情報の例を示す図である。

【図 25】本発明の実施例における、カーソル定義情報取得手段がカーソル定義文生成手段に渡すカーソル定義情報の構成を示す図である。

【図 26】本発明の実施例における、カーソル定義情報取得手段がカーソル定義文生成手段に渡すカーソル定義情報の例を示す図である。

【図 27】本発明の実施例における、出力項目処理情報取得手段が作業変数定義文生成手段と出力処理文生成手段に渡す出力項目情報の構成を示す図である。

【図 28】本発明の実施例における、出力項目処理情報取得手段が作業変数定義文生成手段と出力処理文生成手段に渡す出力項目情報の例を示す図である。

【図 29】本発明の実施例におけるカーソル定義文生成手段の処理内容を示す図である。

【図 30】本発明の実施例におけるカーソル定義文生成手段で使用するカーソル定義文スケルトンを示す図である。

【図 31】本発明の実施例におけるカーソル定義文生成手段によって生成されたカーソル定義文の例を示す図である。

【図 32】本発明の実施例における作業変数定義文生成手段の処理内容を示す図である。

【図 33】本発明の実施例における作業変数定義文生成手段で使用する作業変数定義文スケルトンを示す図である。

【図 34】本発明の実施例における作業変数定義文生成手段によって生成された作業変数定義文の例を示す図である。

【図 35】本発明の実施例における出力処理文生成手段の処理内容を示す図である。

【図 36】本発明の実施例における出力処理文生成手段で使用する出力処理文スケルトンを示す図である。

【図 37】本発明の実施例における出力処理文生成手段によって生成された出力処理文の例を示す図である。

【図 38】本発明の実施例におけるプログラム自動合成手段の処理内容を示す図である。

【図 39】本発明の実施例におけるプログラムスケルトンを示す図である。

【図 40】本発明の実施例におけるプログラム自動合成手段によって生成された出力プログラムの例を示す図である。

【図 41】第 1 の従来技術の構成を示す図である。

【図 42】第 2 の従来技術の構成を示す図である。

【符号の説明】

1 …仕様

2 …生成プログラム

10…データ処理プログラム自動生成システム

101 …仕様入力手段

102 …プログラムスケルトンカスタマイズ情報取得手段

103 …カーソル定義情報取得手段

104 …出力項目処理情報取得手段

105 …カーソル定義文生成手段

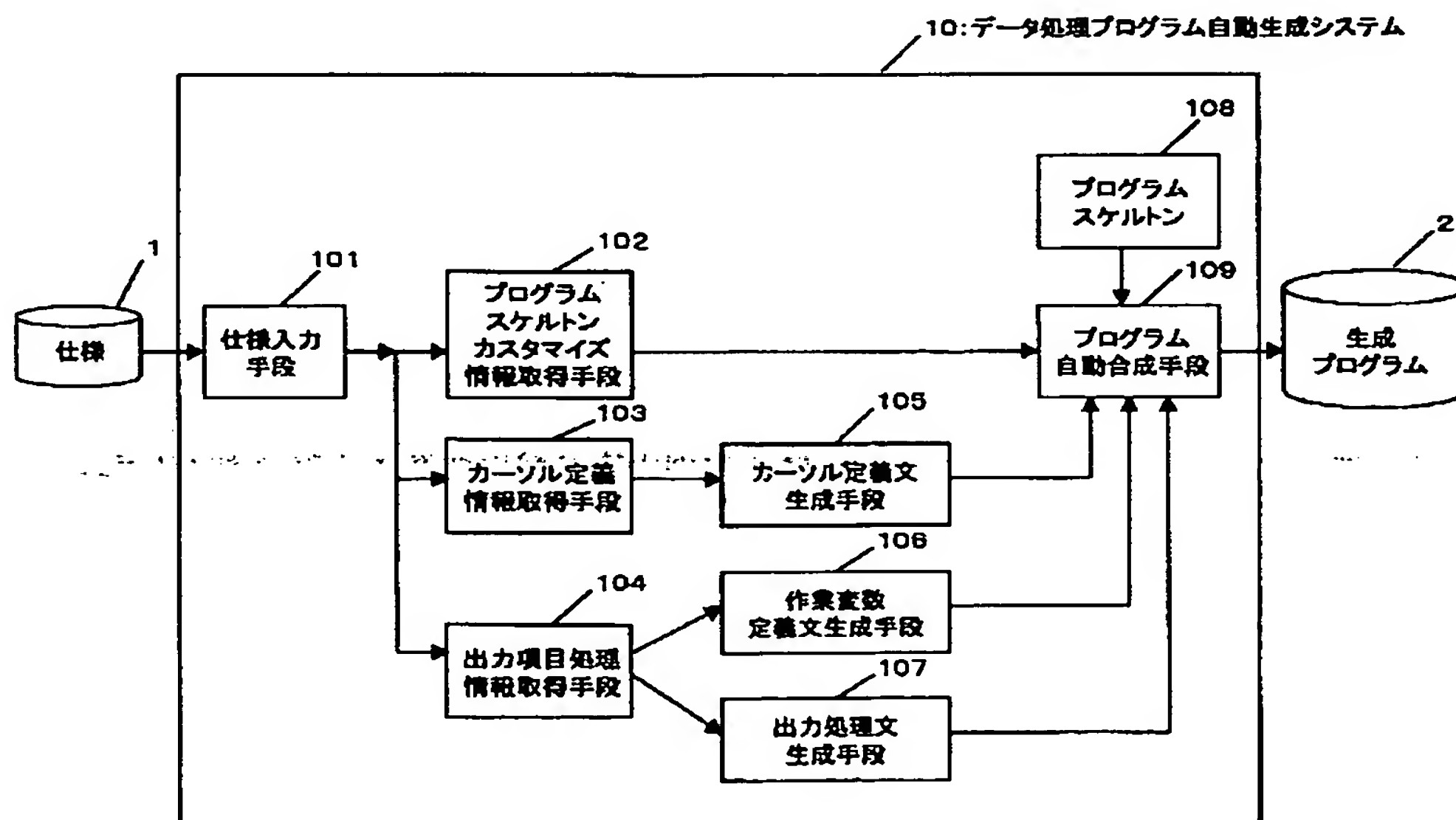
106 …作業変数定義文生成手段

107 …出力処理文生成手段

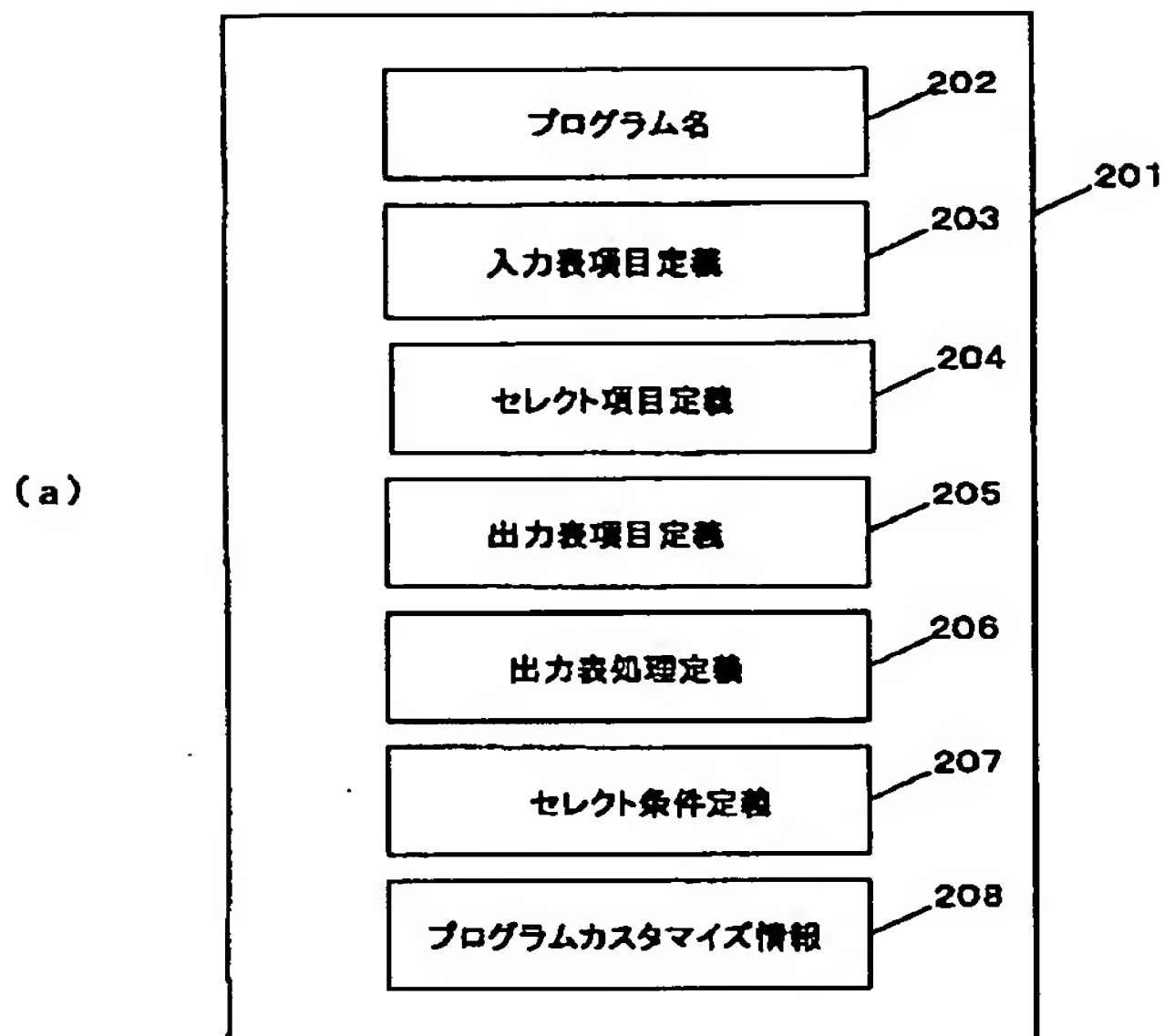
108 …プログラムスケルトン

109 …プログラム自動合成手段

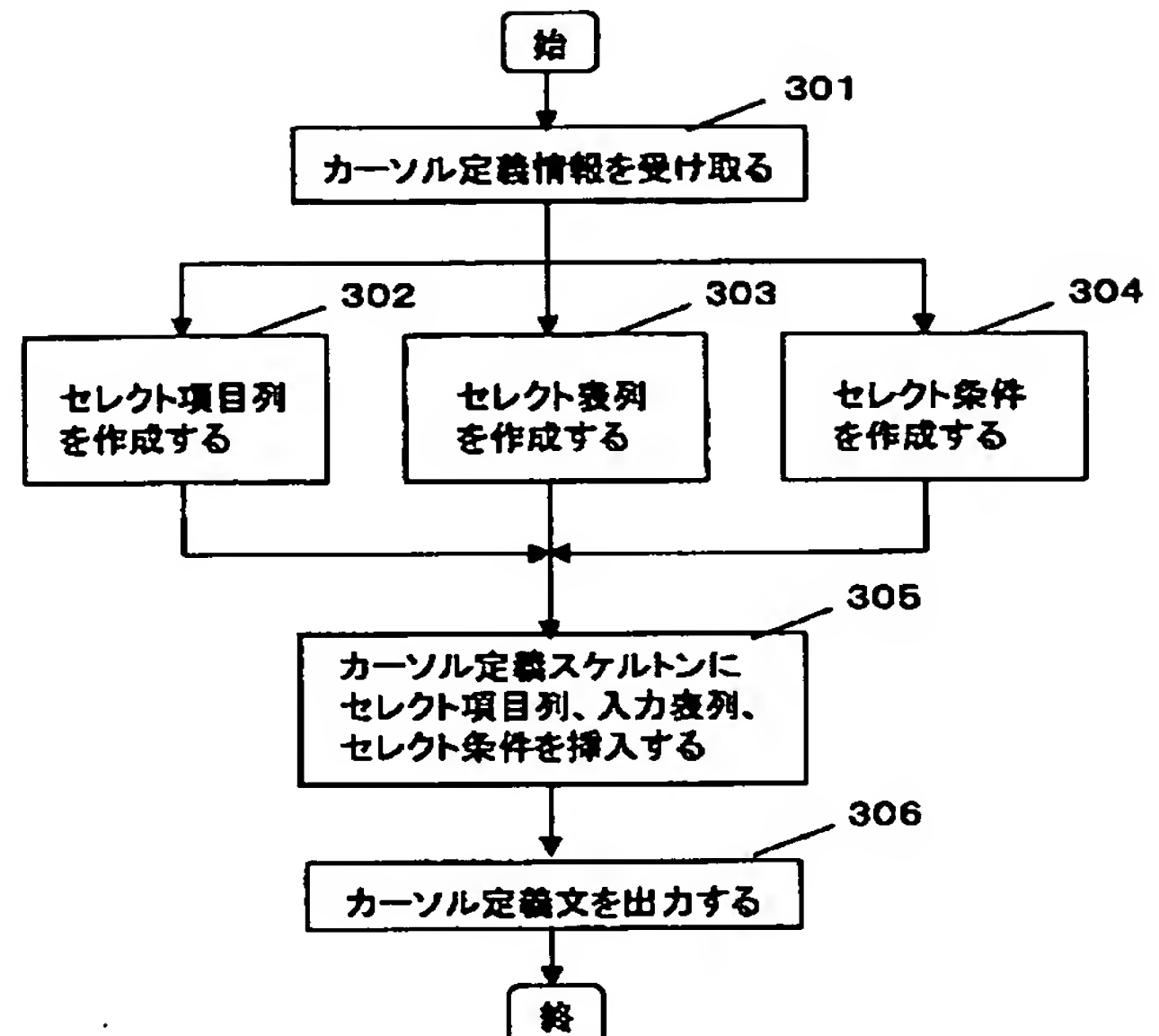
【図 1】



【図 2】



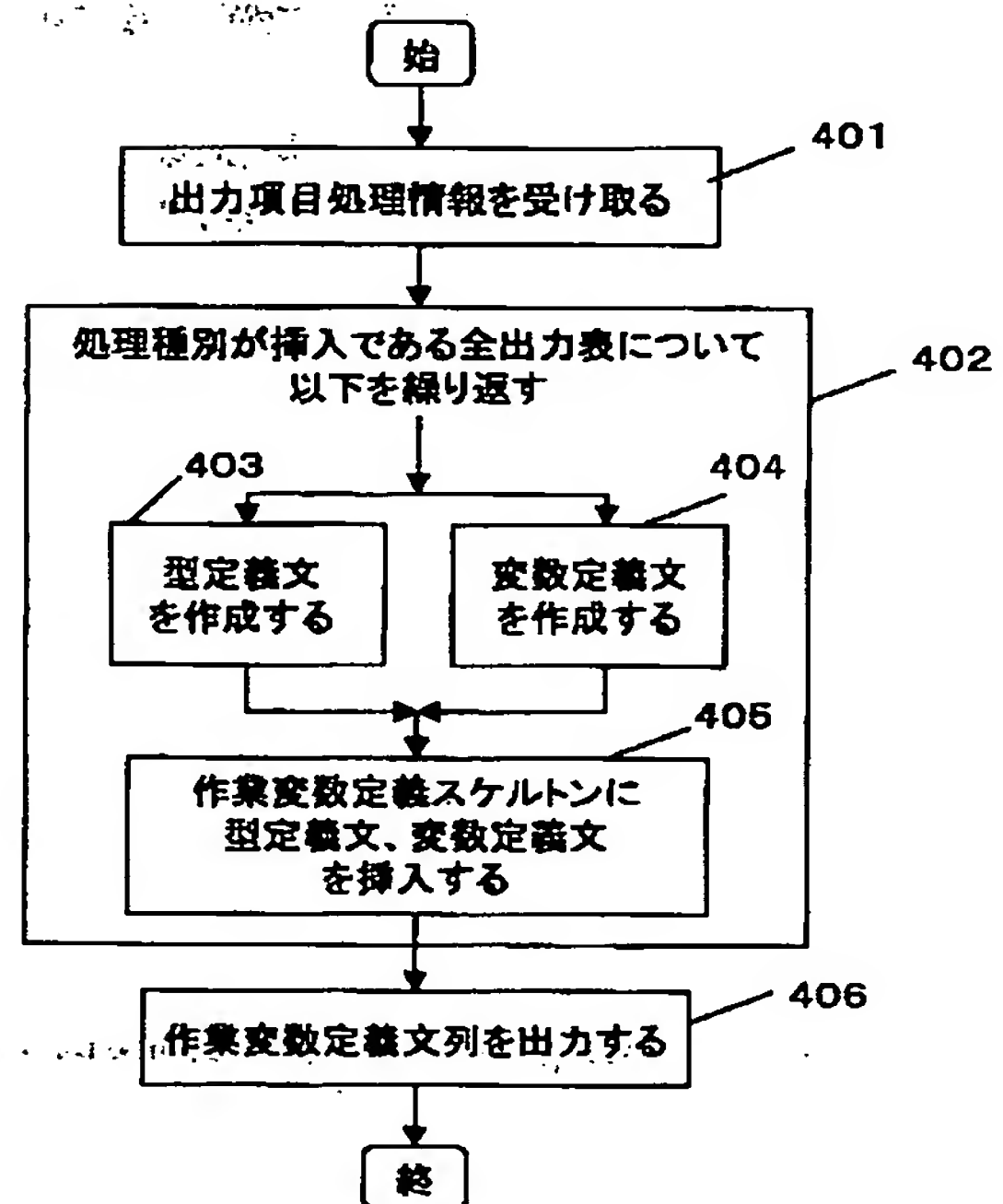
【図 3】



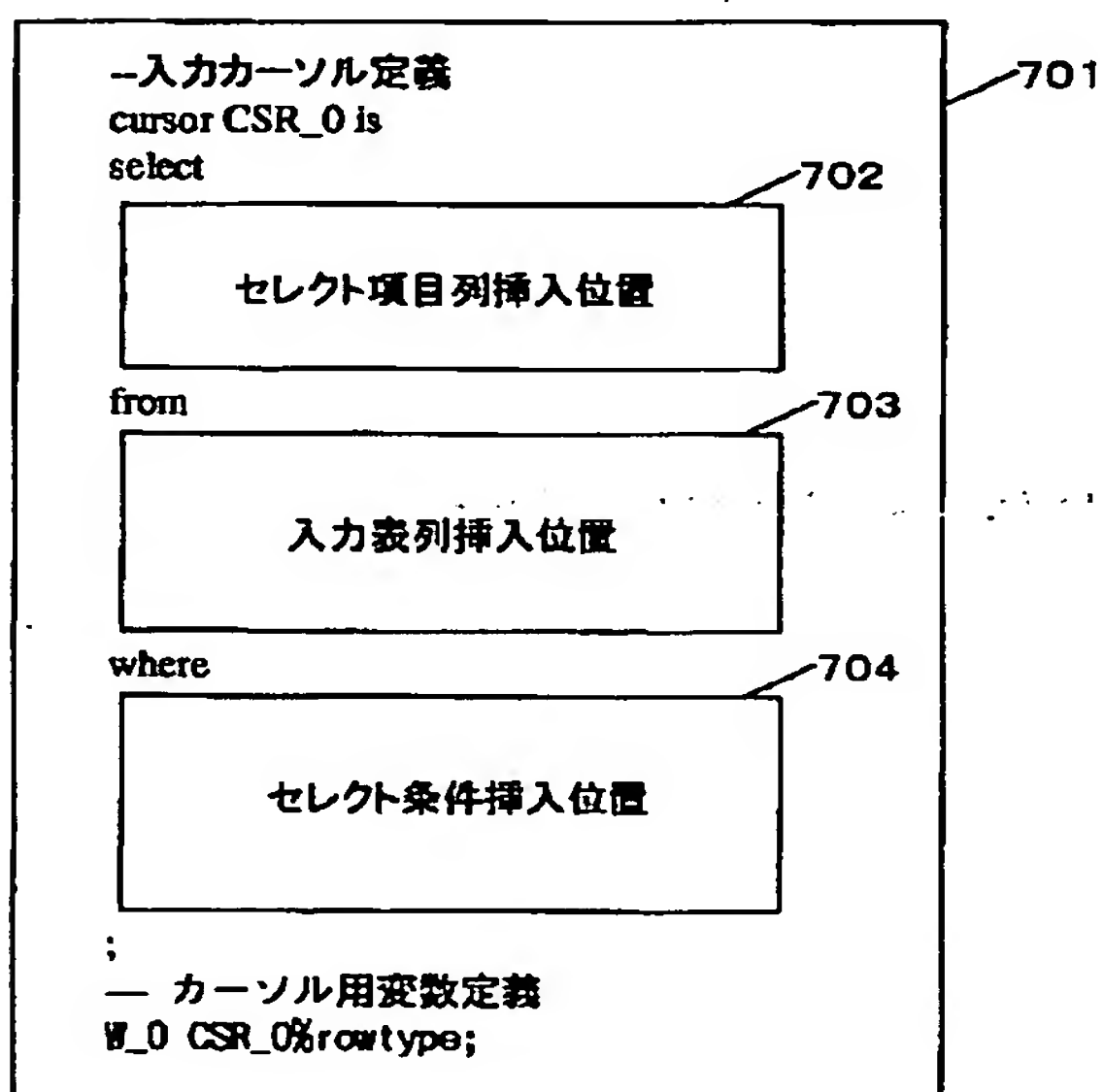
【図 4】

(b)

処理種別	処理条件	更新/削除条件	処理式
挿入	任意	不要	必要
更新	任意	必要	必要
削除	任意	必要	不要



【図 7】

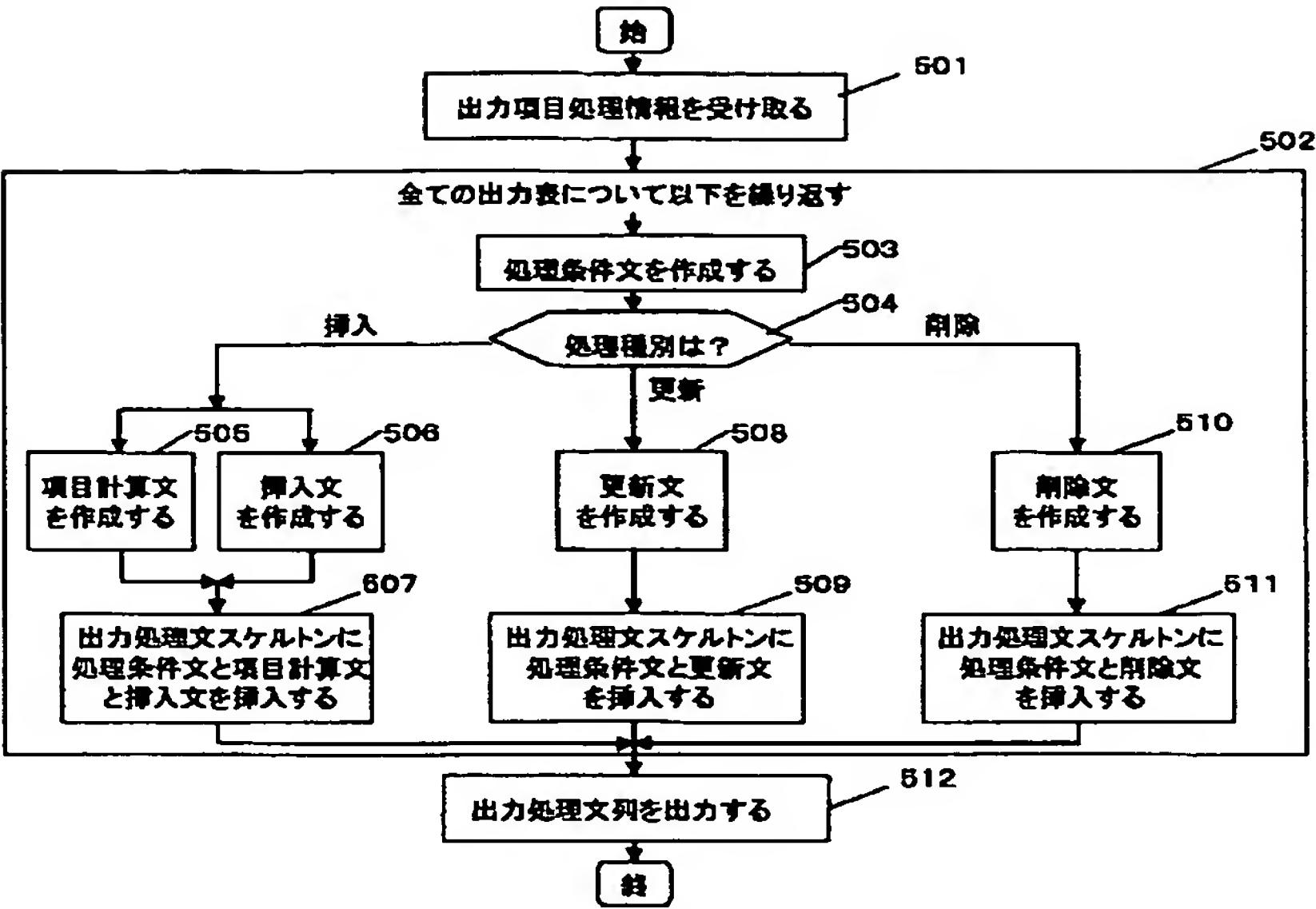


【図 23】

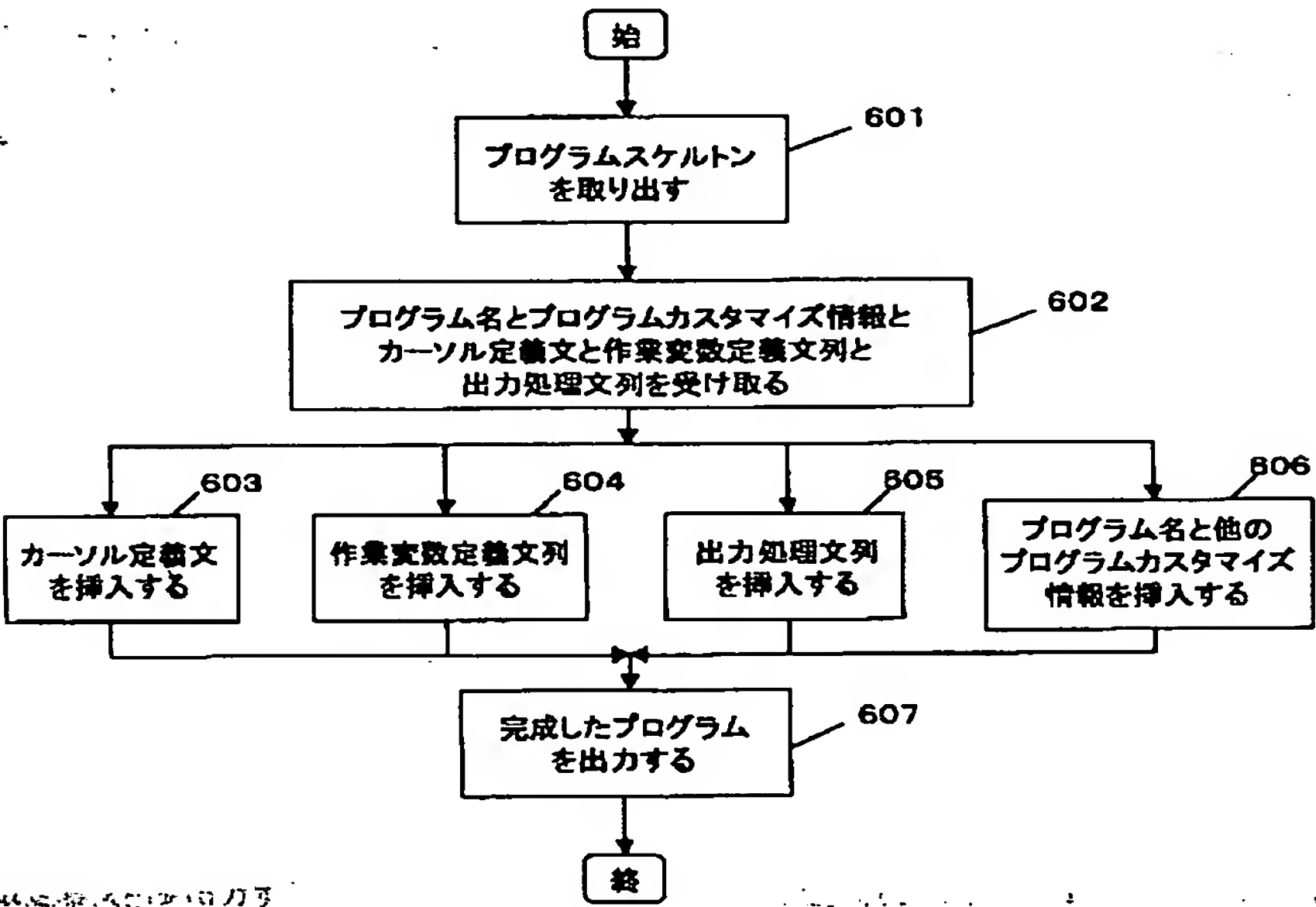
2301: [\*(<<属性名>> : <<属性値>>)\*\*]



【図 5】



【図 6】



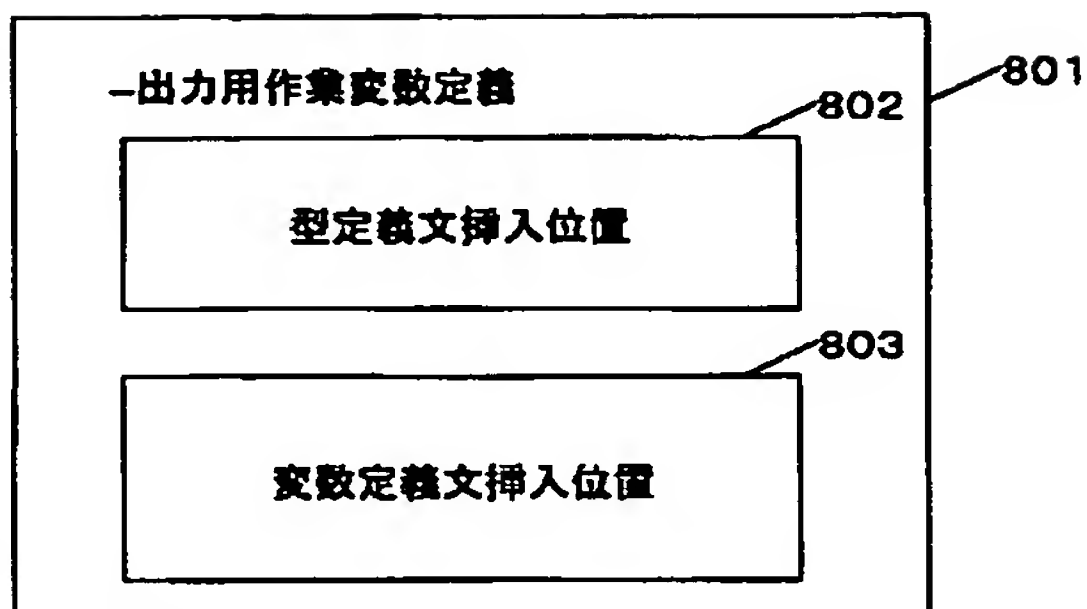
【図 24】

2401:	[ (プログラム名 : "URIAGE_KEISAN")
2402:	( 初期処理 : "Init_module;" )
2403:	( 終了処理 : "End_module;" )
2404:	]

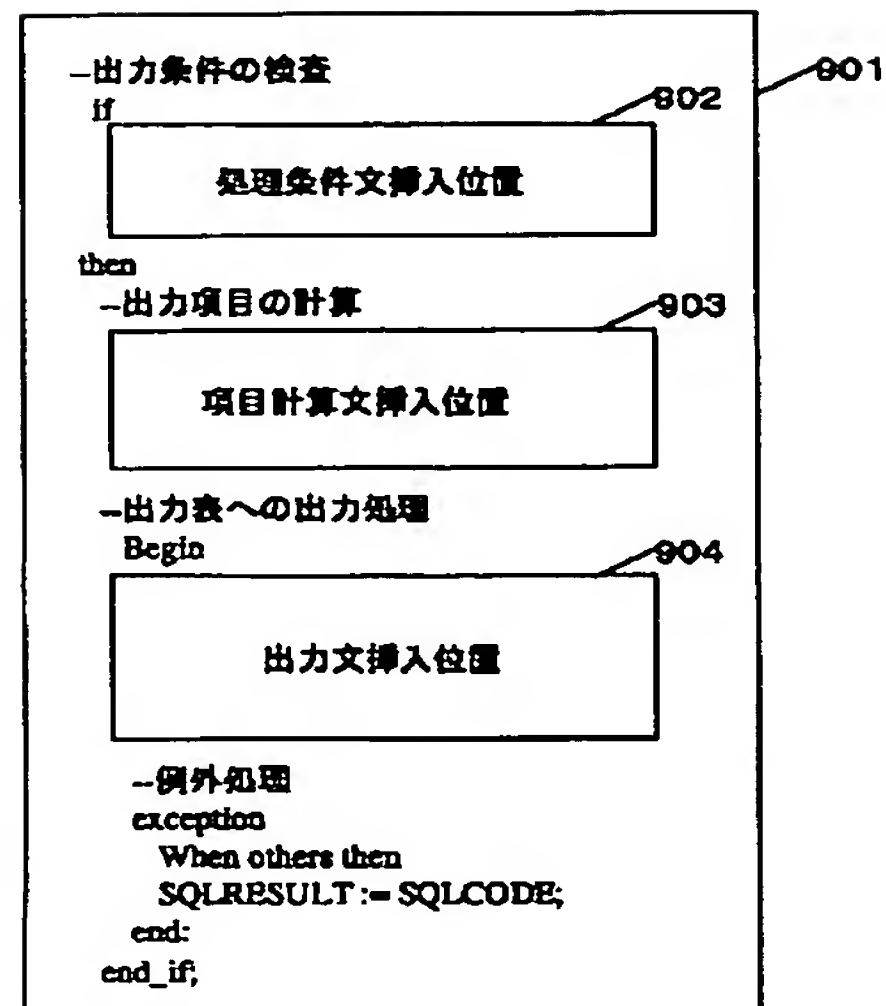
【図 25】

2501:	[ ("入力表" : [(表名 : [項目名]* )]* )
2502:	( "セレクト条件" : 処理条件文字列 )
2503:	( "計算処理" : [計算処理欄の値の文字列]* )
2504:	]

【図8】



【図9】



【図30】

```

3001:  — カーソル定義
3002:  cursor CSR_0 is
3003:  select
3004:    <<セレクト項目列挿入位置>>
3005:  from <<入力表列挿入位置>>
3006:  where <<セレクト条件挿入位置>>
3007:  ;
3008:  — カーソル用変数定義
3009:  W_0 CSR_0%rowtype;

```

(a)

(b)

(c)

(d)

```

insert into <<出力表名>> (
  <<項目名リスト>>
) values (
  <<変数名リスト>>
);

```

```

update <<出力表名>>
set <<更新文リスト>>
WHERE <<更新条件>>;

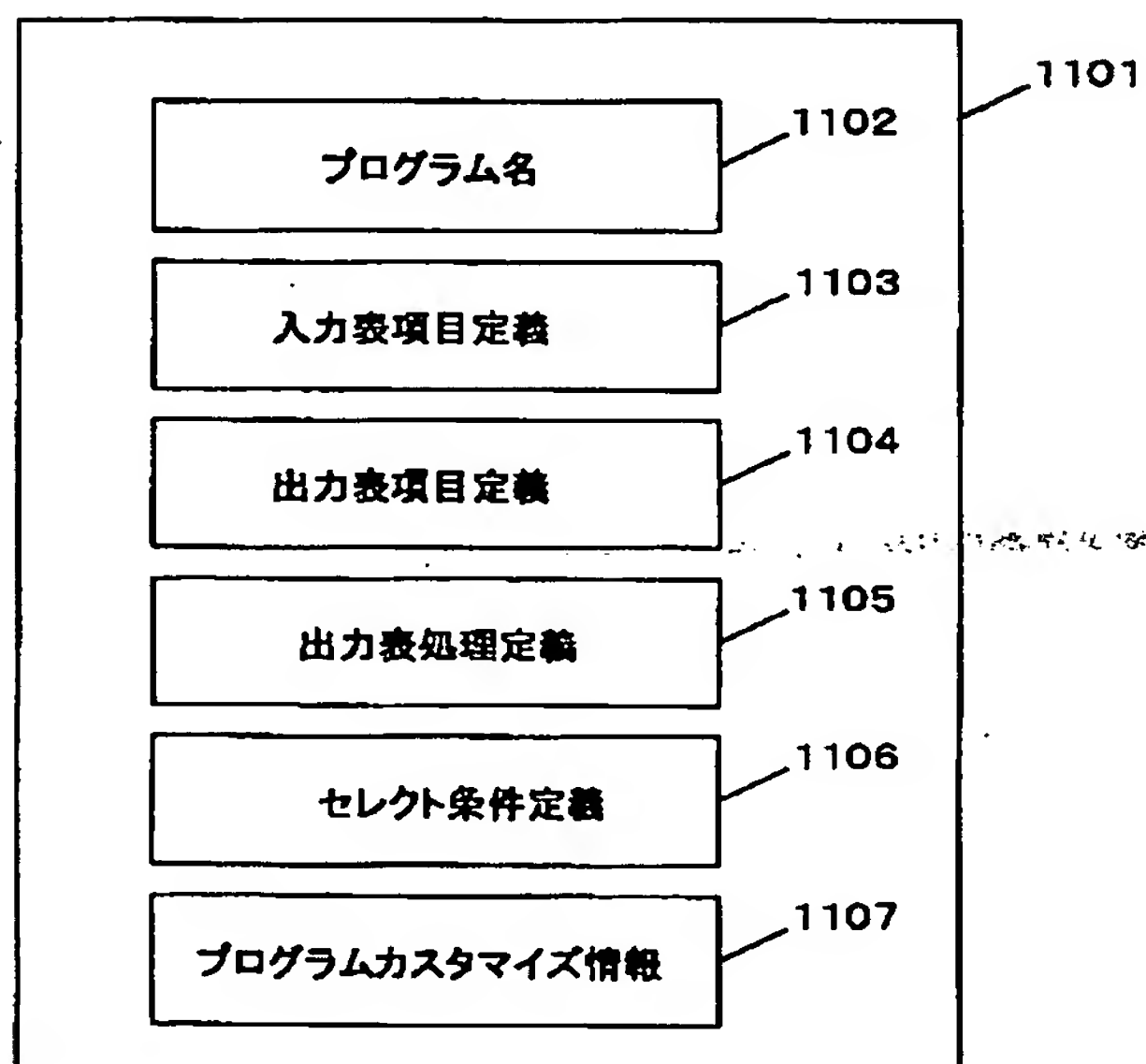
```

```

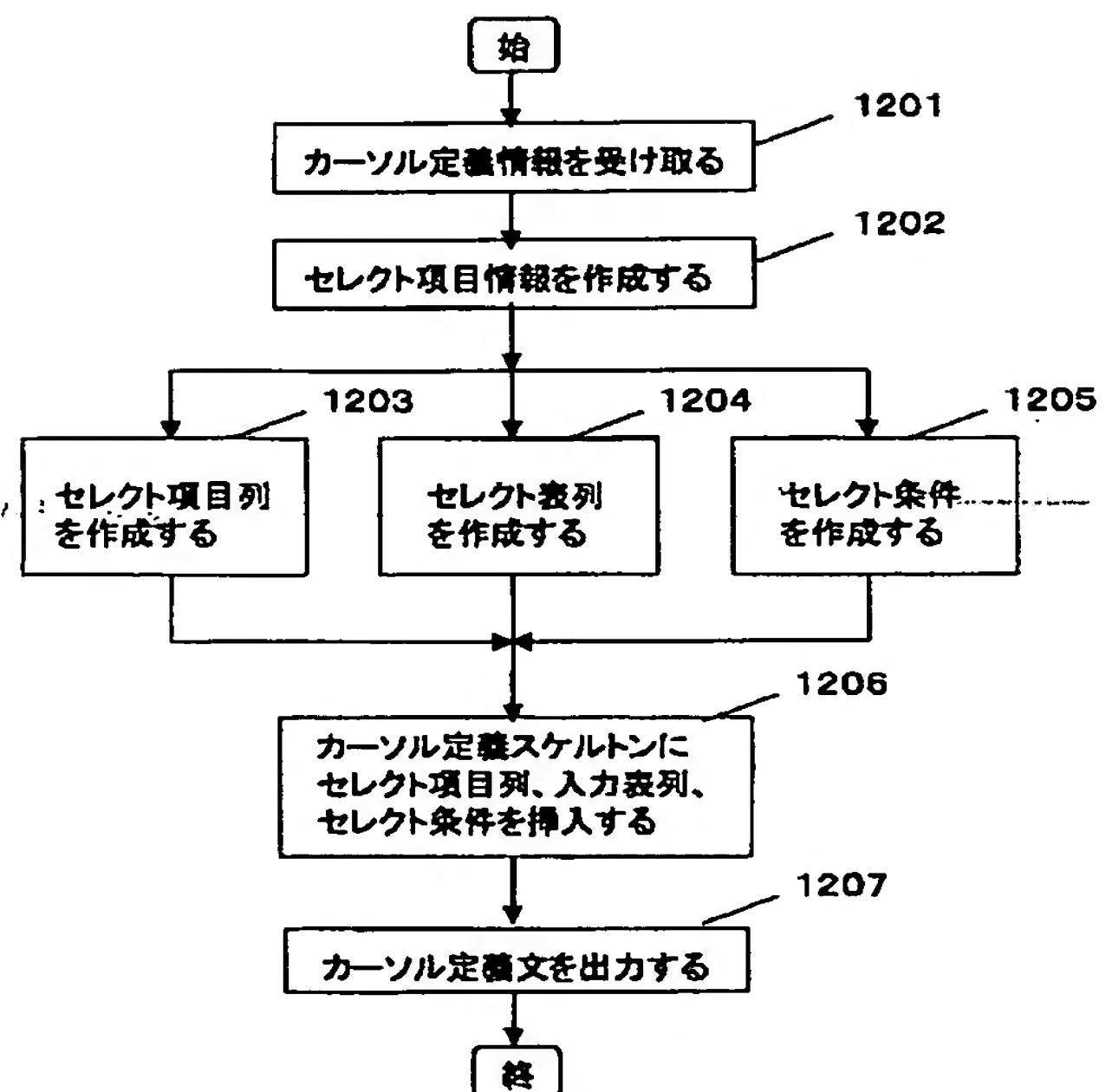
delete from <<出力表名>>
WHERE <<削除条件>>;

```

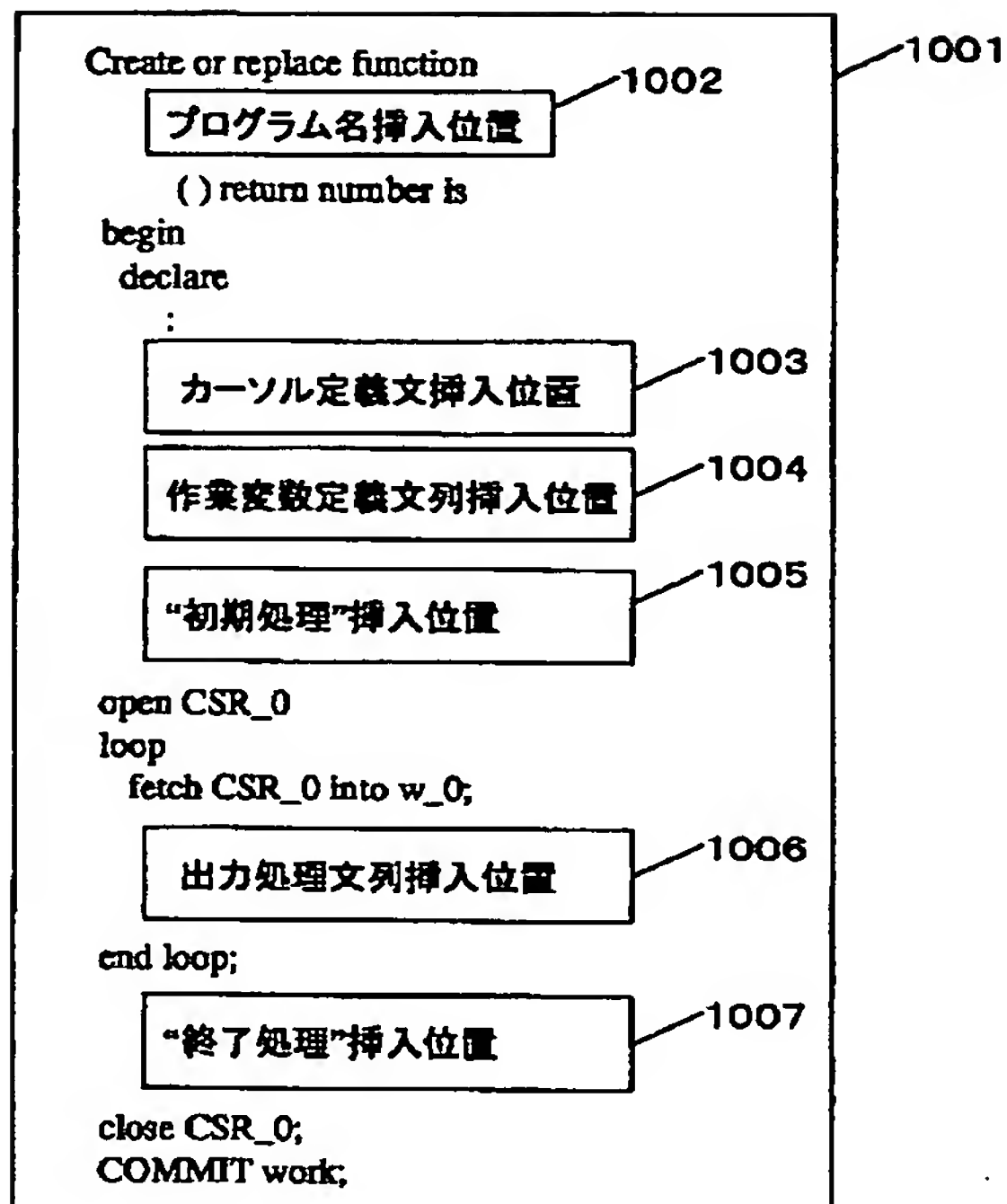
【図11】



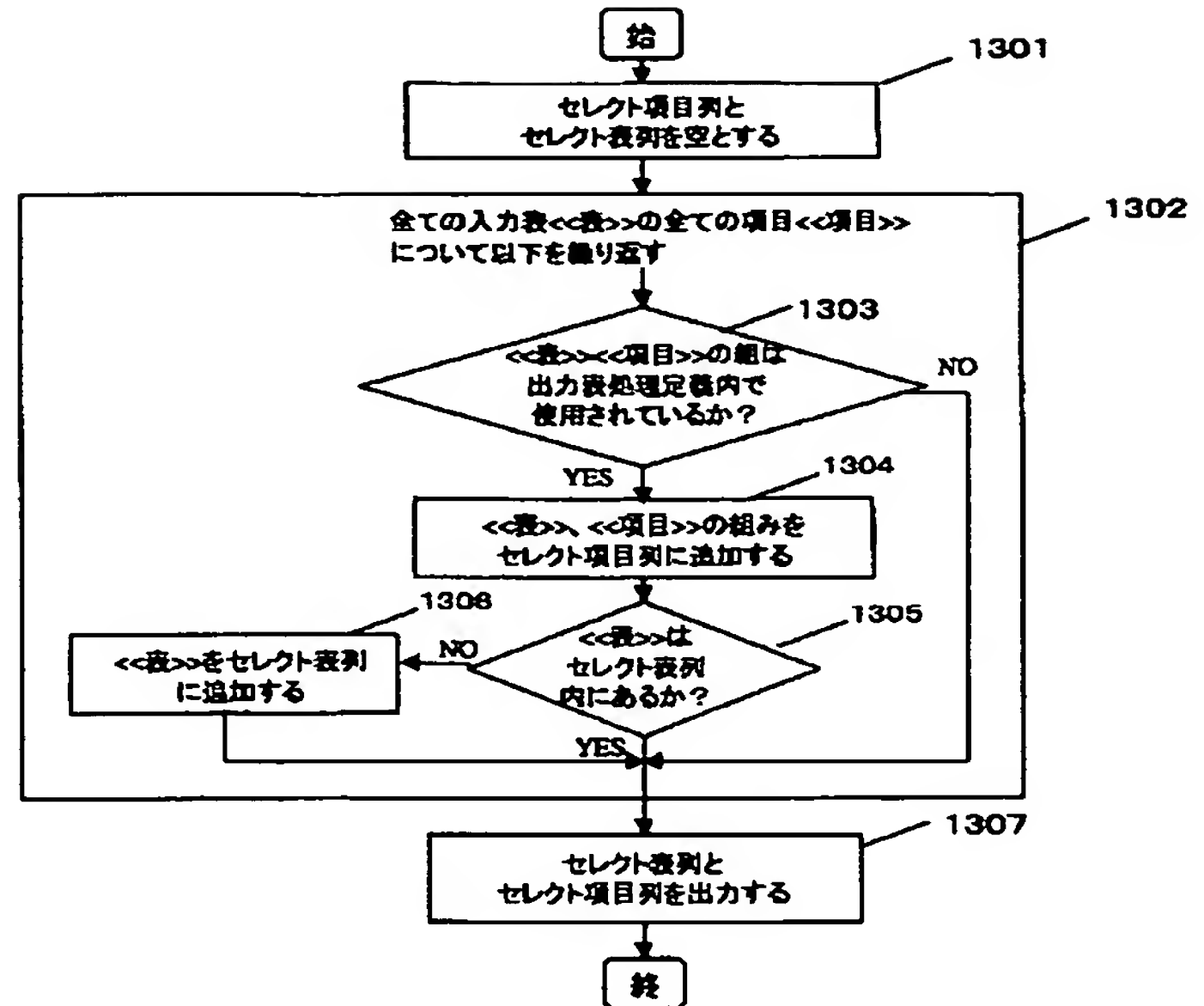
【図12】



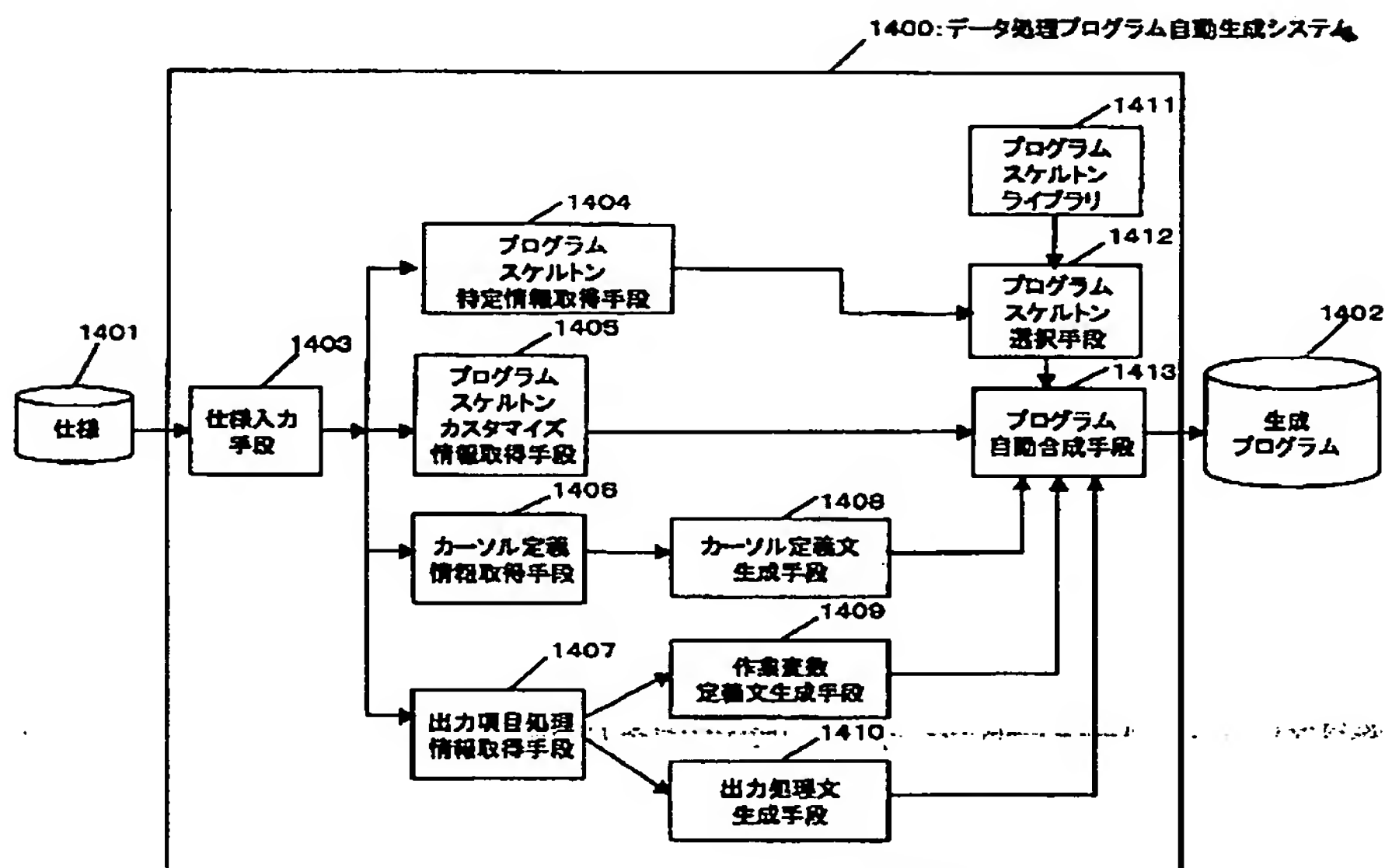
【図10】



【図13】



【図14】

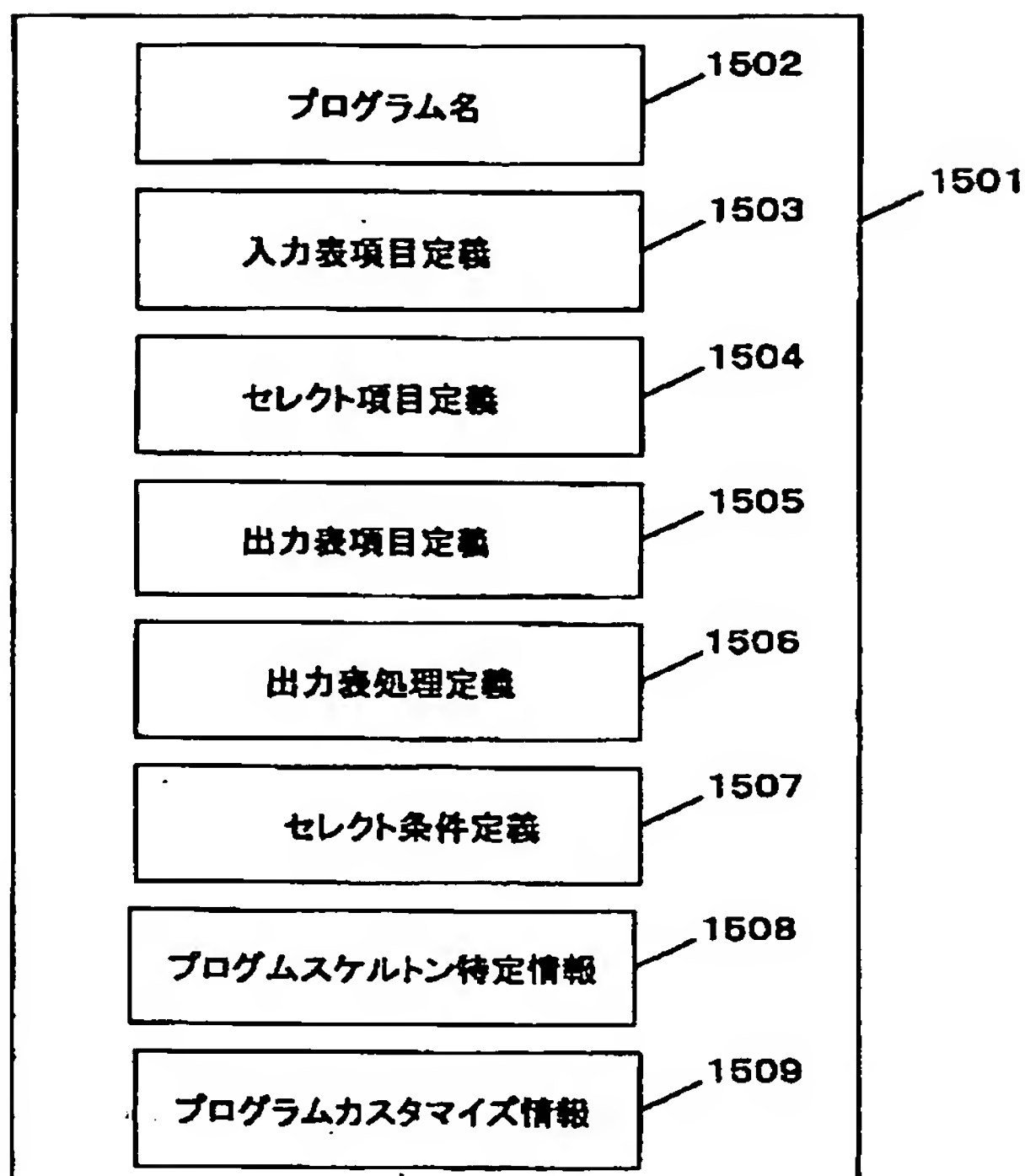


【図27】

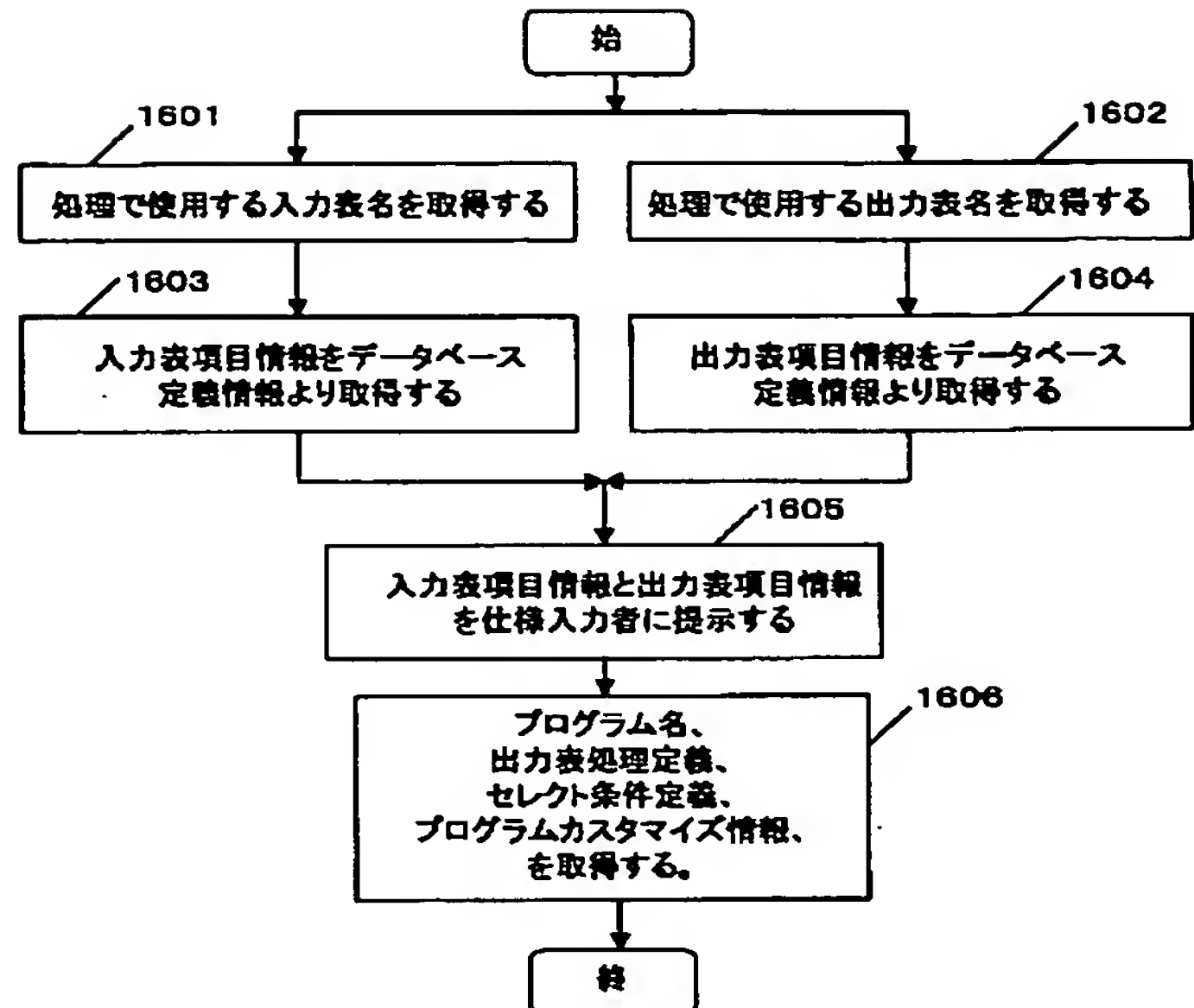
2701: (“出力表” : [(表名 : [(項目名 型 計算処理)]\*)]\*)



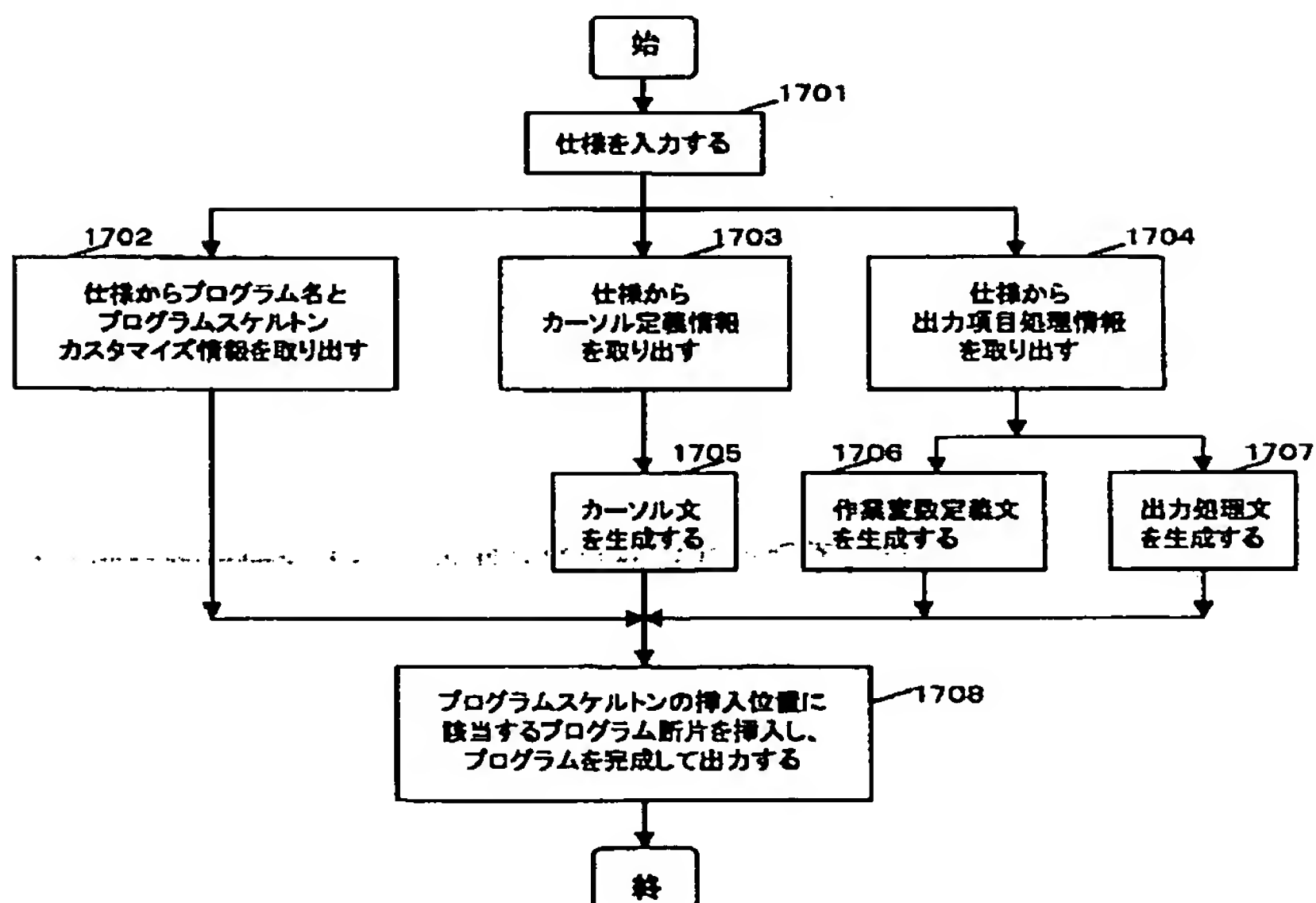
【図 15】



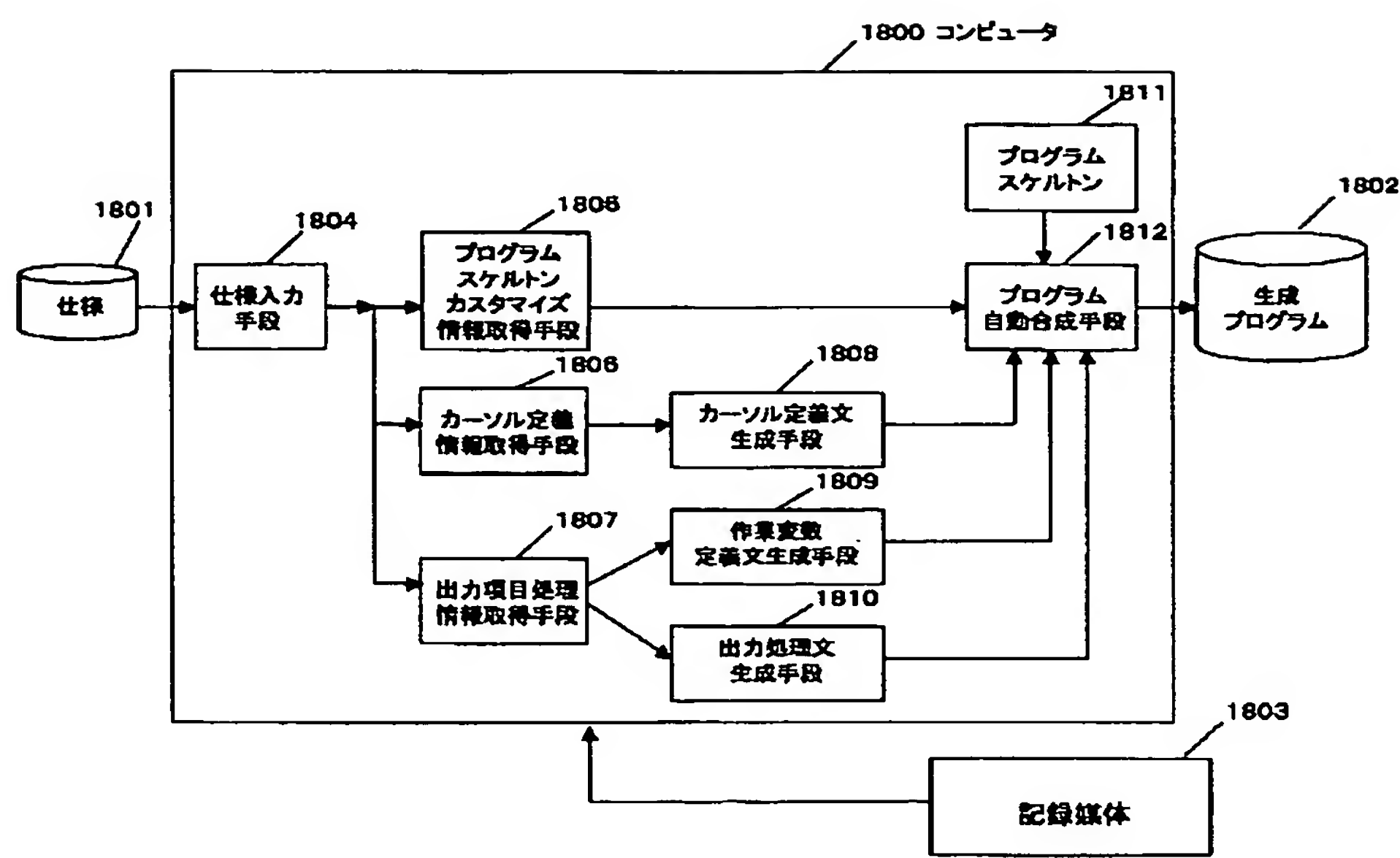
【図 16】



【図 17】



【図18】



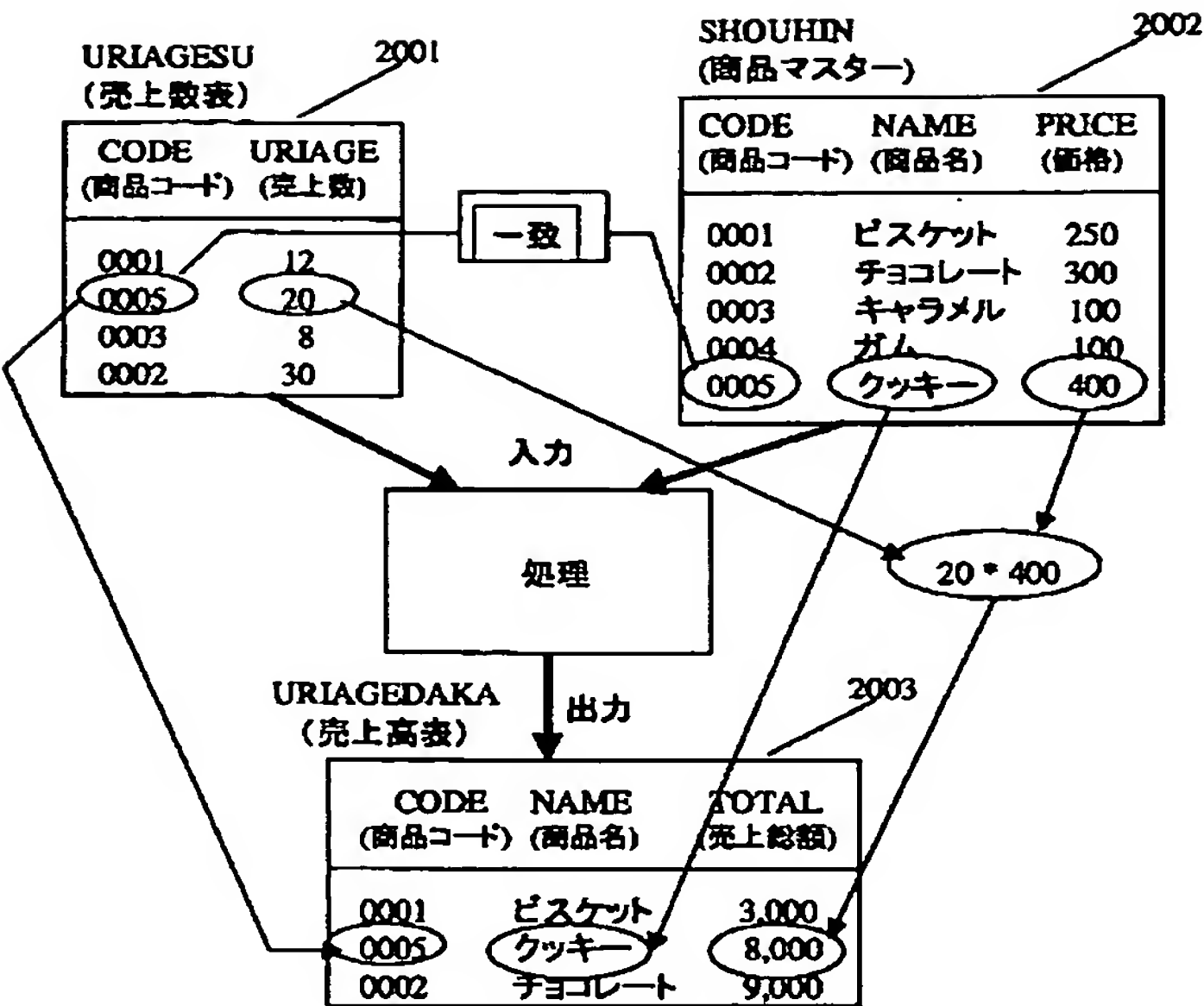
【図19】

プログラム名	URIAGE_KEISAN				1901
表名	入出力区分	項目名	型	計算処理	1902
URIAGESU	入力	CODE	char(4)		1903
		URIAGE	number(6)		1904
SHOUHIN	入力	CODE	char(4)		1905
		NAME	char(13)		1906
		PRICE	number(8)		1907
URIAGEDAKA	出力	CODE	char(4)	URIAGESU.CODE	1908
		NAME	char(13)	SHOUHIN.NAME	1909
		TOTAL	number(14)	URIAGESU.URIAGE * SHOUHIN.PRICE	1910
セレクト条件	URIAGESU.URIAGE > 10 and URIAGESU.CODE = SHOUHIN.CODE				1911
初期処理	Init_module;				1912
終了処理	End_module;				1913

【図33】

3301: 出力用作業変数定義  
3302: type <<作業変数型名挿入位置>> is record(  
3303: W\_ROWID rowId  
3304: <<出力項目定義挿入位置>>  
3305: );  
3306: <<変数定義文挿入位置>>

【図20】



【図31】

```
3101: ー カーソル定義
3102: cursor CSR_0 is
3103: select
3104:   URIAGESU.CODE URIAGESU_CODE,
3105:   URIAGESU.URIAGE URIAGESU_URIAGE,
3106:   SHOUHIN.NAME SHOUHIN_NAME,
3107:   SHOUHIN.PRICE SHOUHIN_PRICE
3108: from URIAGESU, SHOUHIN
3109: where URIAGESU.URIAGE > 10 and
3110:        URIAGESU.CODE = SHOUHIN.CODE
3111: ;
3112: ー カーソル用変数定義
3113: %_0 CSR_0%rowtype;
```

【図21】

```
2101: <<内部情報>>を空とする。
2102: “プログラム名 : <<プログラム名>>”を作成し、<<内部情報>>に追加する
2103: “表 : [”を作成し、<<内部情報>>に追加する
2104: 仕様から各表に関する情報を取得し、各表<<表>>について以下を繰り返す

2105: “( <<表>>, <<入出力区分>>, [”を作成し、<<内部情報>>に追加する
2106: IF <<入出力区分>>は入力?
2107: THEN <<表>>内の各項目<<項目>>について以下を繰り返す
2108:   “(<<項目>>, <<型>>),”を作成し、<<内部情報>>に追加する
2109: ELSE <<表>>内の各項目<<項目>>について以下を繰り返す
2110:   “(<<項目>>, <<型>>, <<計算処理>>),”を作成し、<<内部情報>>に追加する
2111: END_IF
2112: <<内部情報>>の最後の“,”を削除し、“)”を追加する

2113: “処理条件 : <<処理条件>>”を作成し、<<内部情報>>に追加する
2114: その他の属性について以下を繰り返す
2115: “<<属性名>> : <<属性値>>”を作成し、<<内部情報>>に追加する
```



【図22】

```

2201: プログラム名 : URIAGE_KEISAN
2202: 表           : [(URIAGESU, 入力, [(CODE, char(4))
2203:                               (URIAGE, number(6))])
2204:               (SHOUHIN, 入力, [(CODE, char(4))
2205:                               (NAME, char(13))
2206:                               (PRICE, number(8))])
2207:               (URIAGEDAKA, 出力, [(CODE, char(4), "URIAGESU.CODE")
2208:                               (NAME, char(13), "SHOUHIN.NAME")
2209:                               (TOTAL, number(14), "URIAGESU.URIAGE
2210:                               * SHOUHIN.PRICE"))])
2211: セレクト条件 : "URIAGESU.URIAGE > 10 and
2212:               URIAGESU.CODE = SHOUHIN.CODE"
2213: 初期処理    : Init_module;
2214: 終了処理    : End_module;

```

【図26】

```

2601: [(入力表 : (URIAGESU : [CODE URIAGE])
2602:          (SHOUHIN : [CODE NAME PRICE]))
2603:  (セレクト条件 : "URIAGESU.URIAGE > 10 and URIAGESU.CODE = SHOUHIN.CODE")
2604:  (計算処理 : ["URIAGESU.CODE"
2605:              "SHOUHIN.NAME"
2606:              "URIAGESU.URIAGE * SHOUHIN.PRICE"])
2607: ]

```

【図28】

```

2801: (出力表 : [(URIAGEDAKA : [(CODE "char(4)" "URIAGESU.CODE")
2802:                             (NAME "char(13)" "SHOUHIN.NAME")
2803:                             (TOTAL "number(14)" "URIAGESU.URIAGE
2804:                             * SHOUHIN.PRICE")
2805:                             ]
2806:          )])
2807: )

```

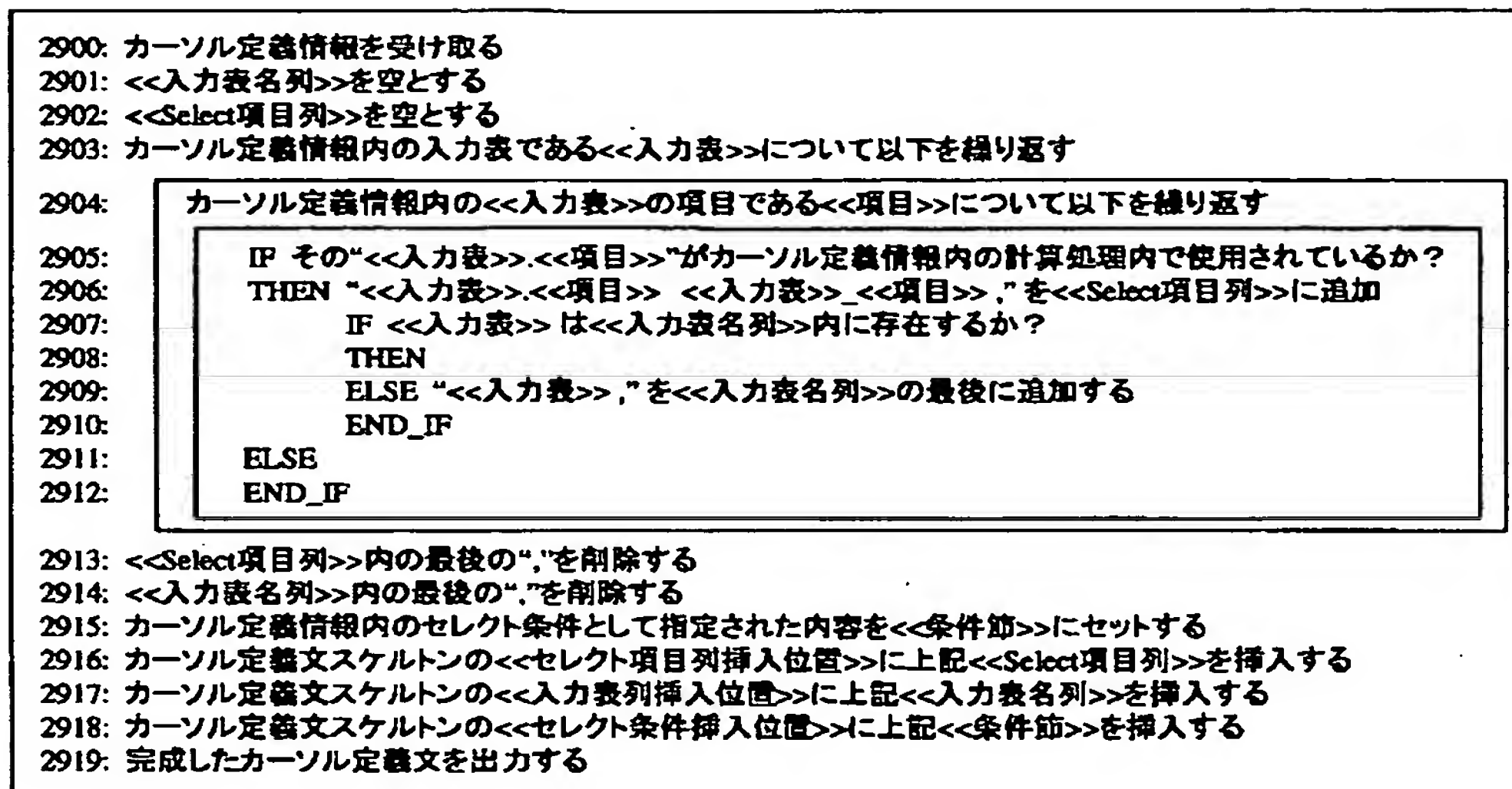
【図34】

```

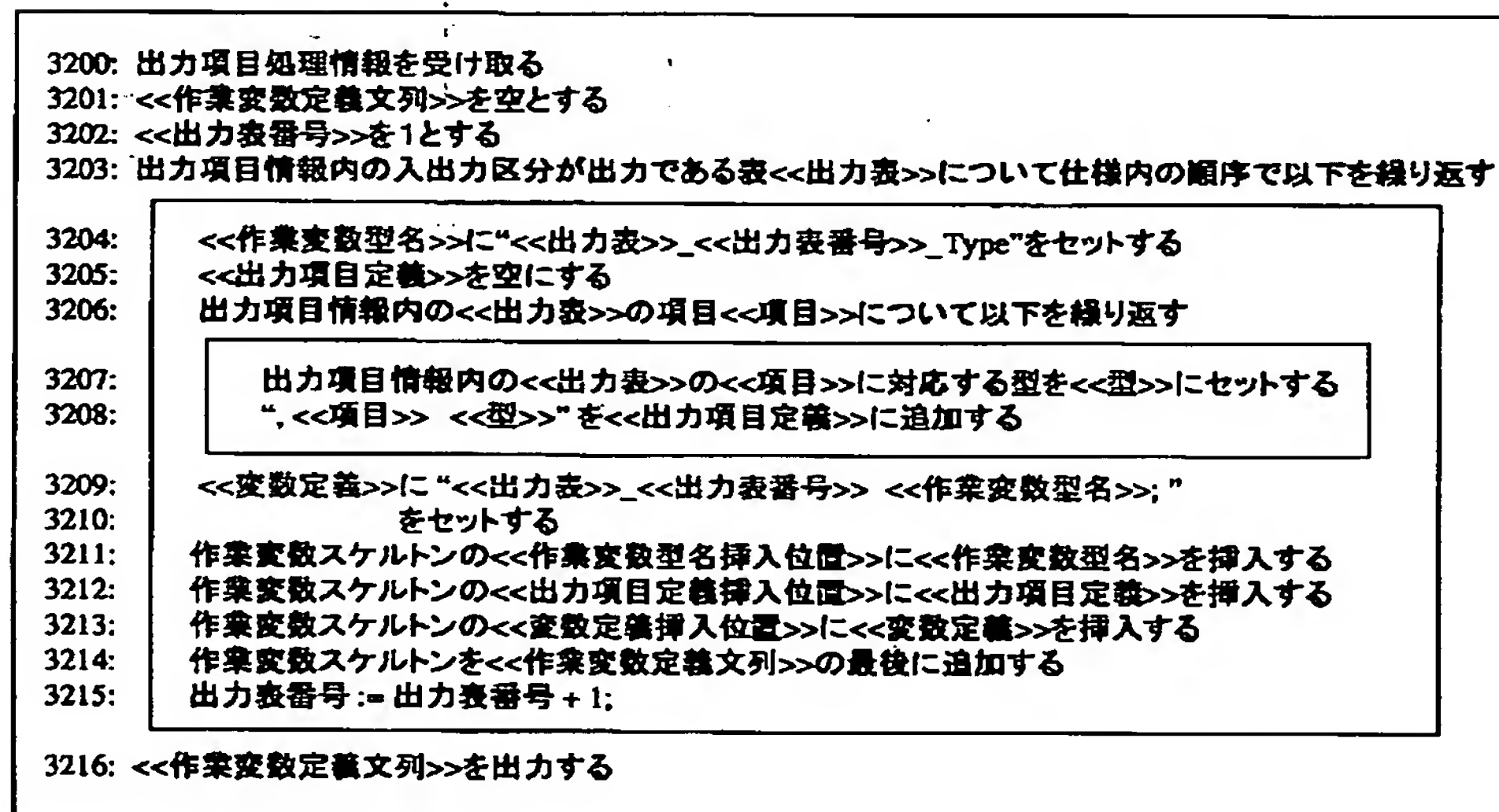
3401: — 出力用変数定義
3402: type URIAGEDAKA_1_Type is record(
3403:   W_ROWID rowid
3404:   , CODE char(4)
3405:   , NAME char(13)
3406:   , TOTAL number(14)
3407: );
3408: URIAGEDAKA_1 URIAGEDAKA_1_Type;

```

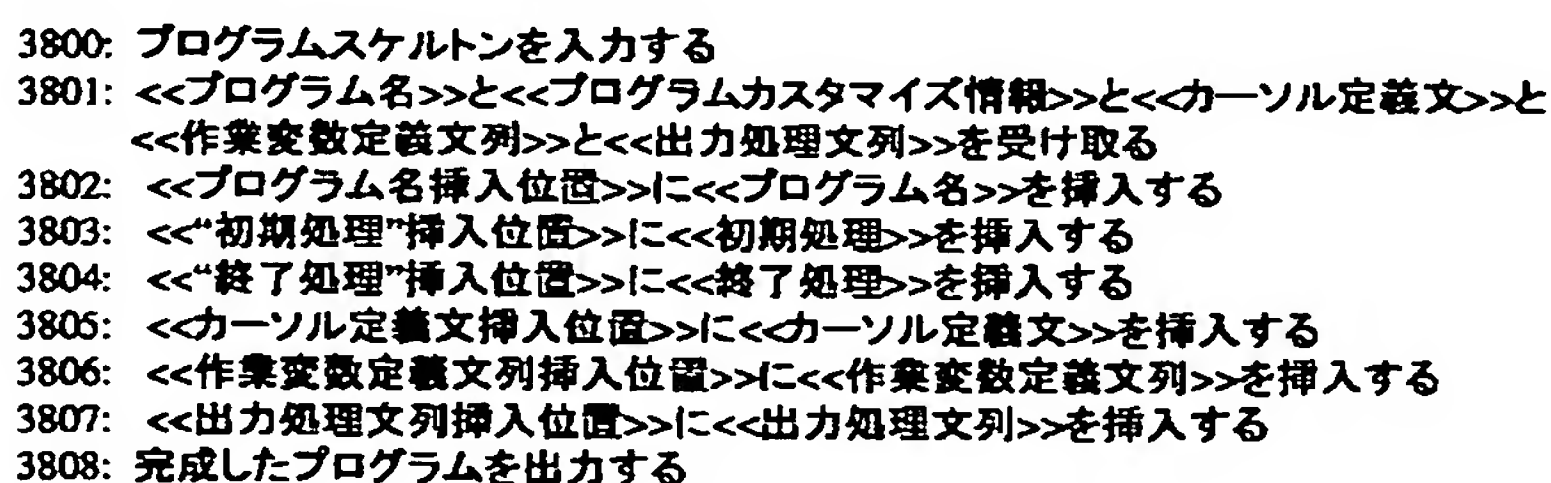
【図29】



【図32】



【図38】



【図35】

```

3500: 出力項目処理情報を受け取る
3501: <<出力処理文列>>を空とする
3502: <<出力表番号>>を1とする
3503: 出力項目処理情報内の入出力区分が出力である表<<出力表>>について仕様の順序で以下を繰り返す

3504:   <<計算文列>>を空とする
3505:   <<項目名列>>を空とする
3506:   <<変数名列>>を空とする
3507:   出力項目処理情報内の<<出力表>>の項目<<項目>>について以下を繰り返す

3508:     出力項目処理情報内の<<出力表>>の<<項目>>に対応する計算処理欄の値のうち、
3509:     “.”を“_”に置換し、表名の前に“W_0.”を追加したものを<<計算式>>にセットする
3510:     “<<出力表>>_<<出力表番号>>.<<項目名>>:=<<計算式>>;”を
3511:     <<計算文列>>の最後に追加する
3512:     “<<項目>>,”を<<項目名列>>の最後に追加する
3513:     “<<出力表>>_<<出力表番号>>.<<項目>>,”を<<変数名列>>の最後に追加する

3514:   <<項目名列>>内の最後の“,”を削除する
3515:   <<変数名列>>内の最後の“,”を削除する
3516:   出力処理文スケルトンの<<項目計算文挿入位置>>に<<計算文列>>を挿入する
3517:   出力処理文スケルトンの<<出力表名挿入位置>>に<<出力表>>を挿入する
3518:   出力処理文スケルトンの<<出力項目名挿入位置>>に<<項目名列>>を挿入する
3519:   出力処理文スケルトンの<<作業変数名挿入位置>>に<<変数名列>>を挿入する
3520:   出力処理文スケルトンを<<出力処理文列>>の最後に追加する
3521:   出力表番号 := 出力表番号 + 1;

3522: <<出力処理文列>>を出力する

```

【図36】

## 出力文スケルトン

```

3601:  — 各出力項目の計算
3602: <<項目計算文挿入位置>>
3603:  — 出力表への出力処理
3604: SQLRESULT := 0;
3605: begin
3606:   insert into <<出力表名挿入位置>>(
3607:     <<出力項目名挿入位置>>
3608:   ) values(
3609:     <<作業変数名挿入位置>>
3610:   );
3611:  — 例外処理
3612: exception
3613: when others then
3614:   SQLRESULT := SQLCODE;
3615: end;

```

【図37】

```

3701:  — 各出力項目の計算
3702: URIAGEDAKA_1.CODE := W_0.URIAGESU_CODE;
3703: URIAGEDAKA_1.NAME := W_0.SHOUHIN_NAME;
3704: URIAGEDAKA_1.TOTAL := W_0.URIAGESU_URIAGE
3705:   * W_0.SHOUHIN_PRICE;
3706:  — 出力表への出力処理
3707: SQLRESULT := 0;
3708: begin
3709:   insert into URIAGEDAKA(
3710:     CODE,
3711:     NAME,
3712:     TOTAL
3713:   ) values(
3714:     URIAGEDAKA_1.CODE,
3715:     URIAGEDAKA_1.NAME,
3716:     URIAGEDAKA_1.TOTAL
3717:   );
3718:  — 例外処理
3719: exception
3720: when others then
3721:   SQLRESULT := SQLCODE;
3722: end;

```



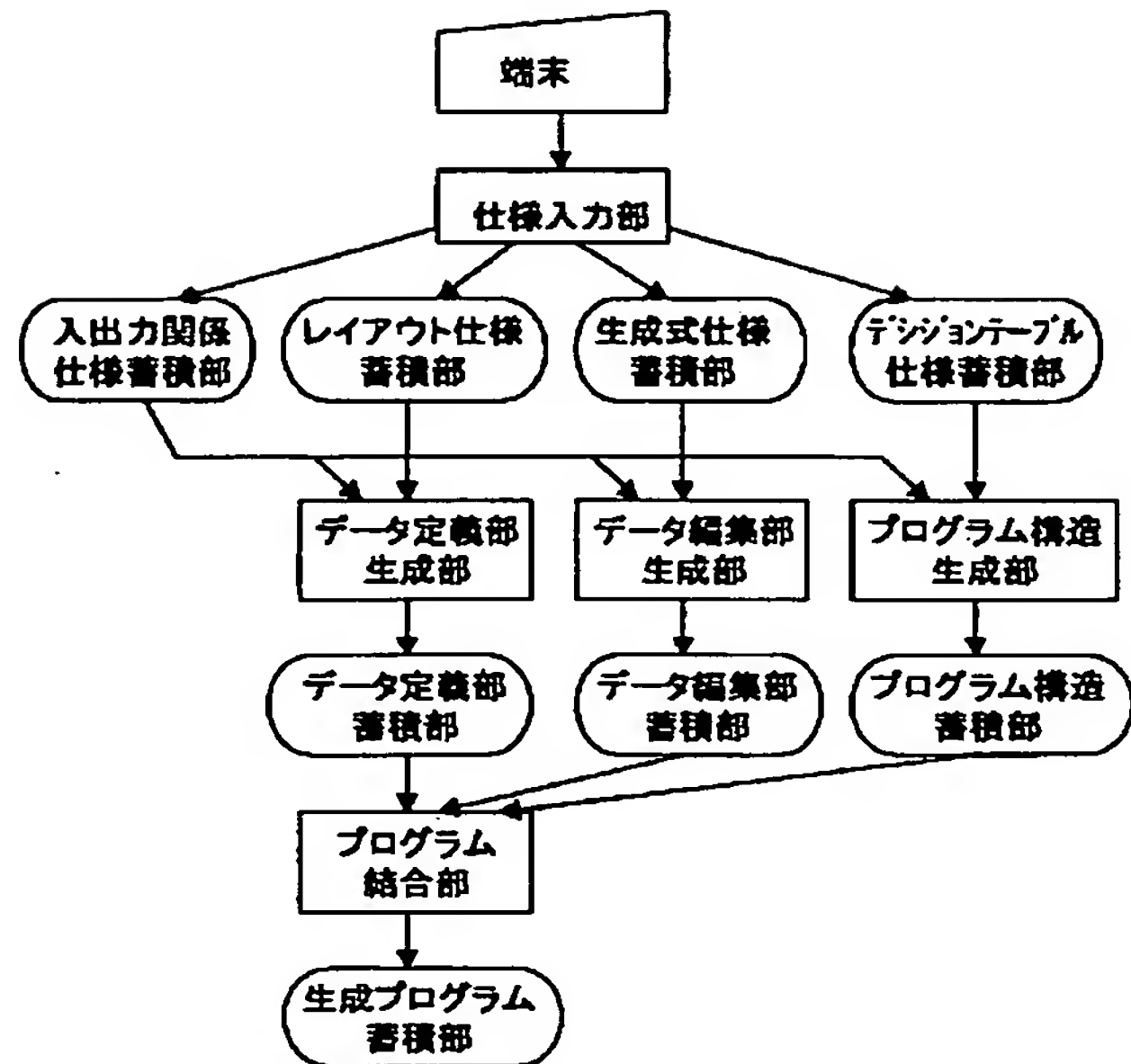
【図39】

```

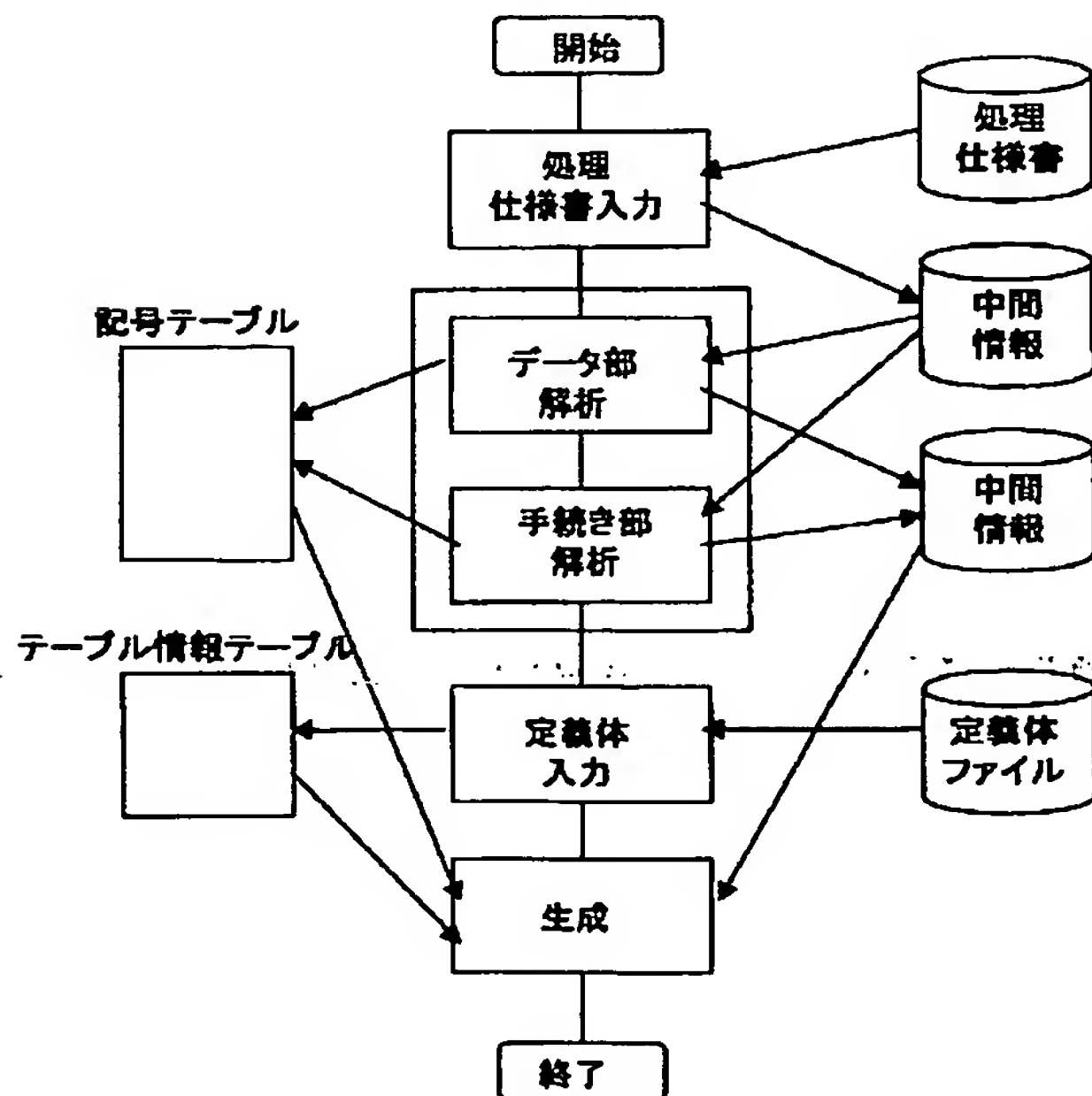
3901: create or replace function <<プログラム名挿入位置>>() return number is
3902: begin
3903:   declare
3904:     /* 宣言部 */
3905:     -- 共通変数定義
3906:     RET_VAL number := 0;
3907:     -- 中間変数定義
3908:     SQLRESULT number := 0;
3909:     <<カーソル定義文挿入位置>>
3910:     <<作業変数定義文挿入位置>>
3911:   begin
3912:     /* 初期処理 */
3913:     -- ユーザ定義の初期処理
3914:     <<初期処理挿入位置>>
3915:     -- 入力表カーソルのオープン
3916:     open CSR_0;
3917:     /* 主処理 */
3918:     -- 結果処理メインブロック(メインループ)
3919:     loop
3920:       -- 入力表カーソル1レコード読み込み
3921:       fetch CSR_0 into E_0;
3922:       -- 入力表のfetch終了時の処理
3923:       if CSR_0%notfound then
3924:         exit;
3925:       end if;
3926:       <<出力処理文挿入位置>>
3927:     end loop;
3928:     /* 終了処理 */
3929:     <<終了処理挿入位置>>
3930:     close CSR_0;
3931:     -- COMMIT処理
3932:     COMMIT WORK;
3933:     return SQLRESULT;
3934:   /* 例外処理 */
3935:   -- 例外処理
3936:   exception
3937:   when others then
3938:     SQLRESULT := SQLCODE;
3939:     ROLLBACK WORK;
3940:     raise;
3941:   end;
3942: end;
3943: end;

```

【図41】



【図42】



【図40】

```

4001: create or replace function URIAGE_NEISAN() return number is
4002: begin
4003:   declare
4004:     /* 宣言部 */
4005:     -- 共通変数定義
4006:     RET_VAL number := 0;
4007:     -- 中間変数定義
4008:     SQLRESULT number := 0;
4009:     -- カーソル定義
4010:     cursor CSR_0 is
4011:     select
4012:       URIAGESU.CODE URIAGESU_CODE,
4013:       URIAGESU.URIAGE URIAGESU_URIAGE,
4014:       SHOUHIN.NAME SHOUHIN_NAME,
4015:       SHOUHIN.PRICE SHOUHIN_PRICE
4016:     from URIAGESU, SHOUHIN
4017:     where URIAGESU.URIAGE > 10 and
4018:           URIAGESU.CODE = SHOUHIN.CODE
4019:   ;
4020:   -- カーソル用変数定義
4021:   W_0_CSR_0%rowtype;
4022:   -- 出力用作業変数定義
4023:   type URIAGEDAKA_1_Type is record(
4024:     W_ROWID rowid
4025:     , CODE char(4)
4026:     , NAME char(13)
4027:     , TOTAL number(14)
4028:   );
4029:   URIAGEDAKA_1 URIAGEDAKA_1_Type;
4030:   begin
4031:     /* 初期処理 */
4032:     -- ユーザ定義の初期処理
4033:     init_module;
4034:     -- 入力表カーソルのオープン
4035:     open CSR_0;
4036:     /* 主処理 */
4037:     -- 繰返処理メインブロック(メインループ)
4038:     loop
4039:       -- 入力表カーソル1レコード読み込み
4040:       fetch CSR_0 into W_0;
4041:       -- 入力表のfetch終了時の処理
4042:       if CSR_0%notfound then
4043:         exit;
4044:       end if;

```

```

4045:       -- 各出力項目の計算
4046:       URIAGEDAKA_1.CODE := W_0.URIAGESU_CODE;
4047:       URIAGEDAKA_1.NAME := W_0.SHOUHIN_NAME;
4048:       URIAGEDAKA_1.TOTAL
4049:         := W_0.URIAGESU_URIAGE * W_0.SHOUHIN_PRICE;
4050:       -- 出力表への出力処理
4051:       SQLRESULT := 0;
4052:       begin
4053:         insert into URIAGEDAKA(
4054:           CODE
4055:           , NAME
4056:           , TOTAL_SALE
4057:         ) values(
4058:           URIAGEDAKA_1.CODE
4059:           , URIAGEDAKA_1.NAME
4060:           , URIAGEDAKA_1.TOTAL
4061:         );
4062:       /* 例外処理 */
4063:       -- 例外処理
4064:       exception
4065:       when others then
4066:         SQLRESULT := SQLCODE;
4067:       end;
4068:     end loop;
4069:     /* 終了処理 */
4070:     End_module;
4071:     close CSR_0;
4072:     -- COMMIT処理
4073:     COMMIT WORK;
4074:     return SQLRESULT;
4075:   /* 例外処理 */
4076:   -- 例外処理
4077:   exception
4078:   when others then
4079:     SQLRESULT := SQLCODE;
4080:     ROLLBACK WORK;
4081:     raise;
4082:   end;
4083: end;

```

**This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record.**

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

☒ **BLACK BORDERS**

☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**

☐ **FADED TEXT OR DRAWING**

☒ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**

☐ **SKEWED/SLANTED IMAGES**

☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**

☐ **GRAY SCALE DOCUMENTS**

☒ **LINES OR MARKS ON ORIGINAL DOCUMENT**

☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**

☐ **OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**